**Vietnam General Confederation of Labor**

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**



**FINAL PROJECT**

**MACHINE LEARNING**

*Instructor*: **Mr. LE ANH CUONG**

*Student*: **Tran Quoc An – 521H0385**

*Class* **: 21H50302**

*Year* **: 23-24**

**HO CHI MINH CITY, 2023**

Vietnam General Confederation of Labor

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**



**FINAL PROJECT**

**MACHINE LEARNING**

*Instructor*: **Mr. LE ANH CUONG**

*Student*: **Tran Quoc An – 521H0385**

*Class* **: 21H50302**

*Year* **: 23-24**

**HO CHI MINH CITY, 2023**

ACKNOWLEDGEMENT

I want to start by sincerely thanking Mr. Le Anh Cuong for his enthusiastic instruction and hard efforts to equip me with the knowledge and abilities I needed to finish this report. He significantly contributed to the development of my mathematical reasoning and expertise.

I am very open to getting comments from teachers so that I may improve my report writing abilities because the impact of the pandemic is too significant for there not to be some flaws in my report.

Finally, I wish you good health and success in your noble career.

*Ho Chi Minh city, 8th December, 2023*

*Author*

*(Sign and write full name)*

*Tran Quoc An*

**THIS PROJECT WAS COMPLETED AT**

**TON DUC THANG UNIVERSIY**

I fully declare that this is my own project and is guided by Mr.Nguyen Quoc Binh; The research contents and results in this topic are honest and have not been published in any form before. The data in the tables for analysis, comments and evaluation are collected by the author himself from different sources, clearly stated in the reference section.

Besides that, the project also uses a number of comments, assessments as well as data from other authors, other agencies and organizations, with citations and source annotations.

**Should any frauds were found, I will take full responsibility for the content of my report.** Ton Duc Thang University is not related to copyright and copyright violations caused by me during the implementation process (if any).

*Ho Chi Minh city, 8th December, 2023*

*Author*

*(Sign and write full name)*

*Tran Quoc An*

SUMMARY

This is a report, I will present about Optimizer in ML, Continual Learning and Test Production to build a model to solve a problem.

Contents

[**A.** **OPTIMIZER IN MACHINE LEARNING** 2](#_Toc154181085)

[**1.** **What is an Optimizer in ML?** 2](#_Toc154181086)

[**2.** **Popular Optimizers in ML** 2](#_Toc154181087)

[**a.** **Gradient Descent (GD)** 2](#_Toc154181088)

[**b.** **Stochastic Gradient Descent (SGD)** 4](#_Toc154181089)

[**c.** **Adagrad (Adaptive Gradient Descent)** 4](#_Toc154181090)

[**d.** **RMS Prop (Root Mean Square)** 5](#_Toc154181091)

[**e.** **Adam (Adaptive Moment Estimation)** 6](#_Toc154181092)

[**B.** **CONTINUAL LEARNING AND TEST PRODUCTION** 7](#_Toc154181093)

[**1.** **Continual Learning** 7](#_Toc154181094)

[**a.** **Definition** 7](#_Toc154181095)

[**b.** **Types of CML:** 7](#_Toc154181096)

[**c.** **CML Process:** 7](#_Toc154181097)

[**d.** **Advantages and limitation of CML** 8](#_Toc154181098)

[**2.** **Test Production** 9](#_Toc154181099)

[**a.** **Introduction** 9](#_Toc154181100)

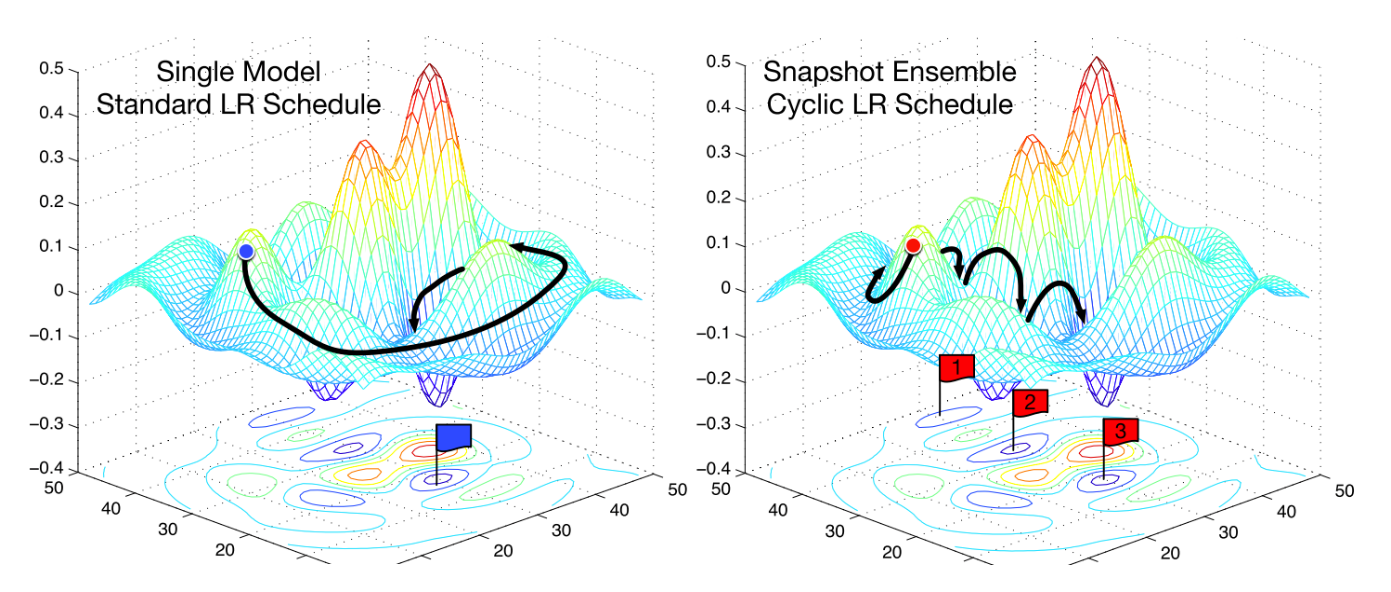
[**b.** **Why Test Production?** 9](#_Toc154181101)

[**c.** **Key concepts in Test Production** 10](#_Toc154181102)

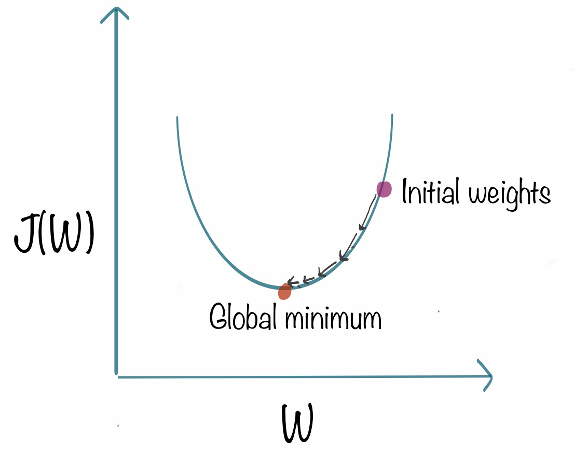
[**d.** **Approachs to Test Production** 10](#_Toc154181103)

[**C.** **Reference** 11](#_Toc154181104)

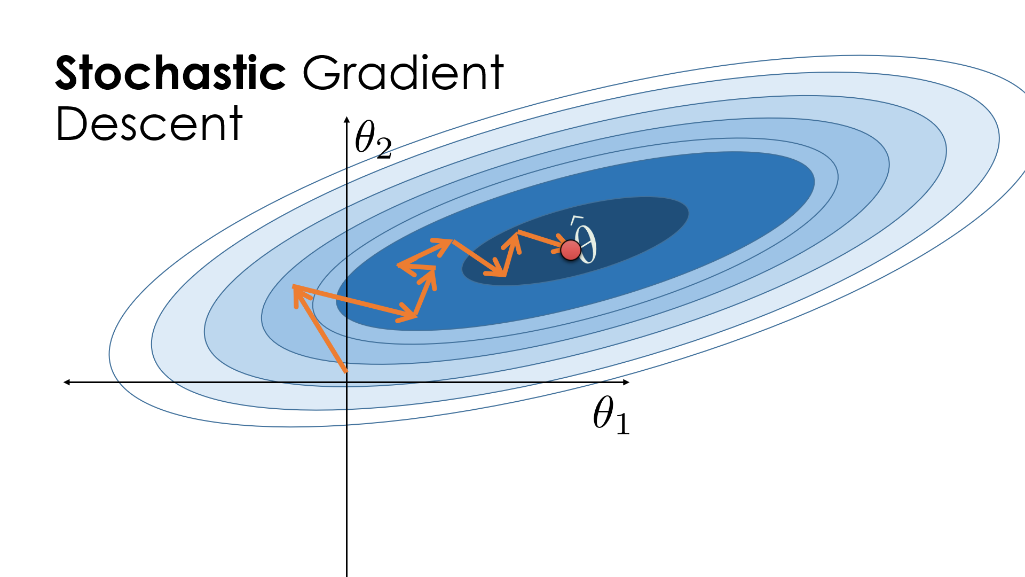
1. **OPTIMIZER IN MACHINE LEARNING**
2. **What is an Optimizer in ML?**

* An optimizer is a crucial part of machine learning that modifies a model's parameters to decrease error or increase learning algorithm performance. Because it optimizes the model's weights and biases depending on the input data, it is essential to the training process.
* An optimizer seeks to determine the ideal combination of parameters that minimizes the loss function, which measures the error between the model's predicted and actual values. The optimizer leads the model towards convergence—where the loss is reduced and the model begins producing correct predictions—by iteratively changing the parameters.
* The process of optimization might be compared to navigating a complicated landscape with many peaks and valleys, where each peak represents a distinct model parameter configuration. Finding the global or local minimum, which denotes the ideal model configuration, is the optimizer's task.

1. **Popular Optimizers in ML**
2. **Gradient Descent (GD)**

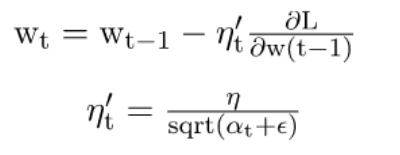
* Gradient Descent is a widely used optimization algorithm that updates the parameters in the direction opposite to the gradient of the loss function. It starts with random initial values for the parameters and updates them iteratively based on the calculated gradients.
* However, it has some downsides too. It is expensive to calculate the gradients if the size of the data is huge. Gradient descent works well for convex functions, but it doesn’t know how far to travel along the gradient for nonconvex functions.

1. **Stochastic Gradient Descent (SGD)**

* The weak of Gradient Descent is expensive to compute when the data is massive. SGD is a variant of Gradient Descent that updates the model’s parameter randomly selected subset of training examples for each iteration.
* SGD has a significant computing benefit over traditional Gradient Descent. SGD evaluates a tiny selection of instances, known as a mini-batch, in each iteration instead of to the full training dataset. Because of this, it is especially helpful for massive data sets that cannot fit in memory.
* SGD computes the loss and gradients based on a mini-batch of training data that is randomly chosen at each iteration. Next, it uses the gradients determined by the mini-batch to update the model's parameters. Through iterative execution of these updates, SGD guides the model in the direction of the loss function minimum.

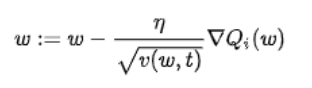
1. **Adagrad (Adaptive Gradient Descent)**

* AdaGrad, or Adaptive Gradient, is a machine learning optimization algorithm that adjusts the learning rate for each parameter based on the historical gradients. It addresses issues like vanishing or exploding gradients by assigning different learning rates to parameters.
* AdaGrad's primary concept is to use the gradients' historical values to determine the optimal learning rate for each parameter. The sum of the squared gradients for each parameter is added up over time as the first step. This accumulation works as an measure for the frequency of parameter updates, assigning greater weight to less often changed parameters and reverse.

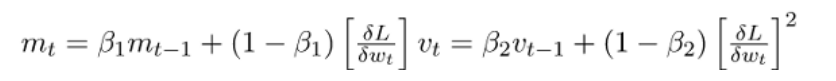


* By dividing the learning rate by the square root of the total sum of squared gradients, AdaGrad makes this modification. The learning rate of a parameter will be higher for parameters with smaller gradients and lower for parameters with larger gradients. By using this method, AdaGrad may adaptably alter the speed of learning for any parameter, so resolving the problem of gradients that vanish or explode.
* AdaGrad's adaptive approach is particularly useful for sparse data, where it prevents excessive updates to non-zero dimensions. However, a drawback is that the learning rate can become overly small over time. Despite its limitations, AdaGrad has influenced the development of subsequent optimization algorithms like RMSprop and Adam.

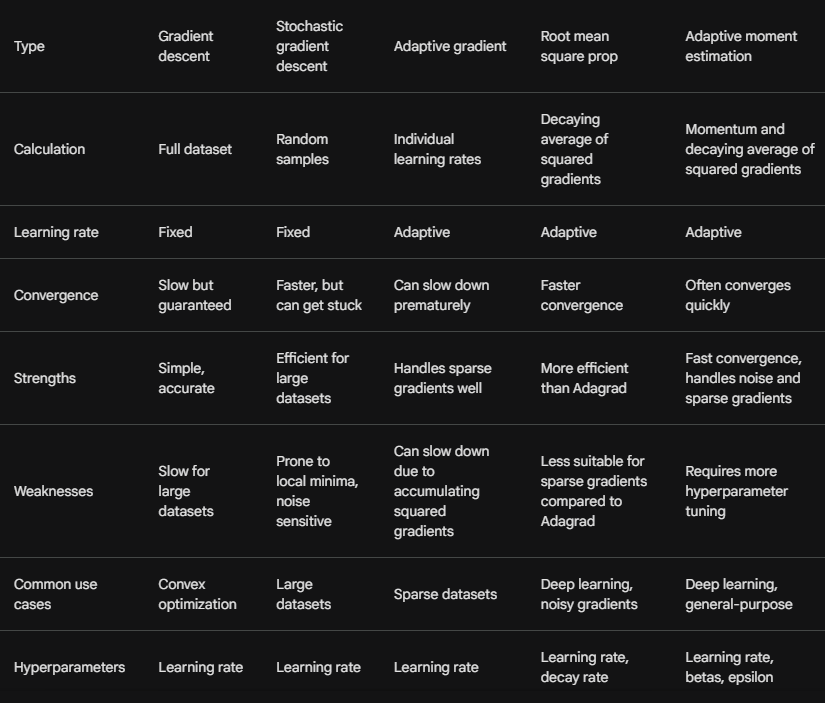
1. **RMS Prop (Root Mean Square)**

* RMSprop (Root Mean Square Propagation) is a machine learning optimization algorithm designed to enhance convergence by dynamically adjusting the learning rate for each parameter
* The main objective of RMSprop is to speed up convergence by adjusting the learning rate based on the moving average of the squared gradients. It divides the learning rate by the square root of the exponentially decaying average of the squared gradients. This normalization prevents the learning rate from getting too large when the gradients are consistently large, and vice versa when the gradients are consistently small.
* The intuition behind RMSprop is that it adapts the learning rate differently for each parameter based on their recent gradient history. Gradients with larger magnitude will result in smaller learning rates, while gradients with smaller magnitude will result in larger learning rates. This adaptive adjustment of the learning rate helps to improve convergence and alleviate the problem of oscillations or divergences that can occur with a fixed learning rate.

1. **Adam (Adaptive Moment Estimation)**

* Adam (Adaptive Moment Estimation) is an optimization algorithm commonly used in machine learning. It combines features of Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propagation (RMSprop) to efficiently update model parameters.
* The Adam optimizer adapts the learning rate for each parameter based on the first-order moments (the mean) and the second-order moments (the uncentered variance) of the gradients. It calculates exponentially decaying averages of past gradients and their square values, effectively estimating the first moment (mean) and the second moment (variance) of the gradients.
* The algorithm uses this estimation of the mean and variance to update the parameters. It combines the benefits of AdaGrad, which scales the learning rate based on the historical gradients, and RMSprop, which adjusts the learning rate adaptively based on the moving average of the squared gradients.

**Comparation between these optimizers**



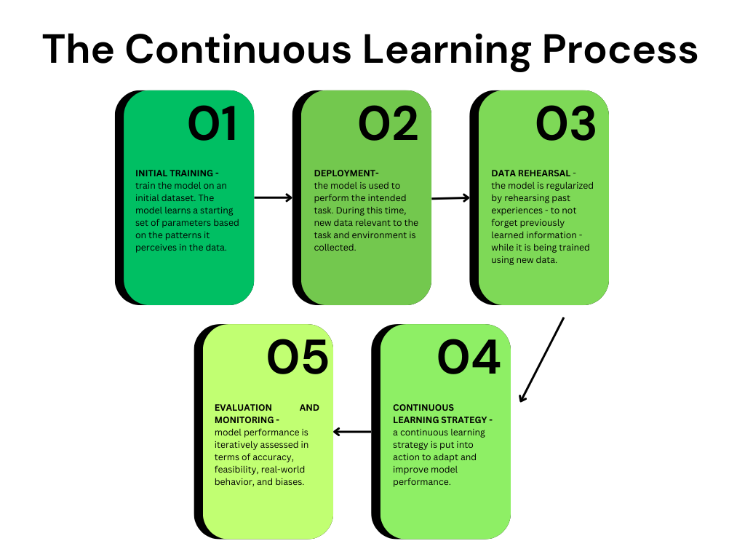
1. **CONTINUAL LEARNING AND TEST PRODUCTION**
2. **Continual Learning**
3. **Definition**

* Continual Learning is a process in which a model learns from new data streams without being re-trained
* Continuous learning models iteratively update their parameters to reflect new distributions in the data, in contrast to traditional approaches where models are trained on a static dataset, deployed, and then regularly re-trained.
* Through learning from the most recent iteration and updating its knowledge when new data becomes available, the model enhances itself in the latter phase. Because of their intrinsic dynamic nature, models with a continuous learning life-cycle can stay relevant throughout time.

1. **Types of CML:**

* There are multiple continuous ML approaches to modeling. Popular strategies include incremental learning, transfer learning and lifelong learning

1. **CML Process:**

* It is an evolution of traditional ML modeling, so it involves many of the same modeling principle: re-processing, model selection, hyperparameter tuning, training, deployment and monitoring.
* However, CML have 2 additional steps that are ‘data rehearsal’ and ‘the implementation of a continuous learning strategy’. These processes serve to ensure that the model is learning from streams of new data efficiently based on the application and context of the data task.

1. **Advantages and limitation of CML**

**Advantages:**

* Continuous learning can be flexible for all types of data projects: descriptive, diagnostic, predictive, and prescriptive.
* **Generalization**. Continuous learning empowers the model to be more robust and accurate in the face of new data.
* **Retention of information**. By employing a continuous learning strategy, the model considers previous knowledge gained in past iterations, enabling it to accumulate information over time.
* **Adaptability**. A model employing continuous learning adapts to new knowledge – such as concept drift and new trends – thereby having greater predictive capabilities in the long run.

**Limitations**:

* **Cost**. Continuous learning approaches, while effective, also tend to be more computationally complex than traditional ones as the model needs to consistently adapt to new data. Said complexity often translates into higher economic costs because it necessitates more data, human, and computing resources.
* **Model management**. Every time a model’s parameters update based on new data, a new model is formed. Therefore, a continuous learning approach may generate a large number of models, complicating the identification of best-performing ones.
* **Data drift**. For a continuous learning approach to be worthwhile, we must process a large volume of new data. However, such a model risks the chance of losing predictive capabilities if the feature distribution changes abruptly. Learn more about [data drift](https://www.datacamp.com/tutorial/understanding-data-drift-model-drift) in a separate article.

1. **Test Production**
2. **Introduction**

* Testing in machine learning (ML) production environments goes beyond standard software testing. It delves into the unique aspects of models and their behavior in real-world scenarios. This explanation will introduce through the various facets of test production in ML, including its purposes, key concepts, and different testing approaches.

1. **Why Test Production?**
   * Machine learning production testing is the process of ensuring the reliability and performance of an ML system in a real-world environment
   * Deploying an ML model into production is exciting, but it's crucial to ensure its robustness and effectiveness. Test production helps in:

* Identifying real-world performance: Training data might not fully represent production data. Testing evaluates the model's performance on real-world inputs, detecting biases, performance degrades, or unexpected behavior.
* Monitoring model drift: Over time, the data distribution can shift, causing the model's performance to decline. Monitoring tracks these changes and triggers alerts for retraining or intervention.
* Ensuring stability and reliability: Production systems demand reliability. Testing guarantees the model makes consistent predictions, handles errors gracefully, and doesn't introduce unexpected downtime.
* Building trust and confidence: Robust testing fosters trust in the model's capabilities and encourages stakeholders to embrace its outputs.

1. **Key concepts in Test Production**

* **Training-serving skew:** This occurs when the data distribution in production differs from the training data. Testing helps identify such discrepancies and mitigate their impact.
* **Model drift:** The gradual shift in the data distribution over time can cause the model's performance to drift. Monitoring helps detect and address this drift.
* **Latency and scalability:** Production models need to respond quickly and handle high volumes of requests. Testing measures latency and ensures the model scales efficiently.
* **Explainability and fairness:** Understanding the model's reasoning and identifying potential biases is crucial for responsible AI. Testing helps explore these aspects.

1. **Approachs to Test Production**

* **Batch testing**: Evaluating the model on a representative batch of real-world data before deployment.
* **Online testing:** Continuously monitoring the model's performance in production using live data streams.
* **Shadow testing:** Running the model in parallel with the existing system without affecting production, comparing predictions and identifying discrepancies.
* **Canary testing:** Deploying the new model to a small subset of users first to monitor performance and gather feedback before full rollout.
* **Adversarial testing:** Evaluating the model's robustness against malicious attacks intended to manipulate its predictions.
* Remember: Test production is an ongoing process, not a one-time effort. Continuously monitoring and refining your testing approaches as the model and data evolve is crucial for maintaining its optimal performance in a dynamic production environment.

1. **Reference**
2. Dubose, E. (2023, November 17). *What Is Optimizer In Machine Learning*. Robots.net. Retrieved December 22, 2023, from <https://robots.net/fintech/what-is-optimizer-in-machine-learning/>
3. Gupta, A. (2023, September 13). *Optimizers in Deep Learning: A Comprehensive Guide*. Analytics Vidhya. Retrieved December 22, 2023, from https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on-deep-learning-optimizers/