**Vietnam General Confederation of Labor**

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**



**FINAL PROJECT**

**MACHINE LEARNING**

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

*Student*: **PHAM TRAN TIEN PHAT – 521H0285**

*Class* **: 21H50302**

*Year* **: 25**

**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*: **PHAM TRAN TIEN PHAT – 521H0285**

*Class* **: 21H50302**

*Year* **: 25**

**HO CHI MINH CITY, 2023**

ACKNOWLEDGEMENT

I would like to express my sincere thanks to Mr. Le Anh Cuong for imparting so much knowledge to me so that I can complete this exercise. And I want to say thanks to Ton Duc Thang University, when this place gives me an opportunity to follow up with large exercises so that I can consolidate the knowledge learned through the lectures.

**THIS PROJECT WAS COMPLETED AT**

**TON DUC THANG UNIVERSIY**

I fully declare that this is my own project and is guided by Mr. Le Anh Cuong; 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 few comments, assessments as well as data from other authors, other agencies, and organizations, with citations and source annotations.

**Should any frauds be 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, 19th December 2023*

*Author*

*(Sign and write full name)*

*phat*

*Pham Tran Tien Phat*

CONFIRMATION AND ASSESSMENT SECTION

**Instructor confirmation section**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Ho Chi Minh 2023*

*(Sign and write full name)*

**Evaluation section for grading instructor**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Ho Chi Minh 2023*

*(Sign and write full name)*

SUMMARY

Enhancing the training of a model necessitates a good grasp of machine learning optimizers. Various widely recognized algorithms such as Adam, Stochastic Gradient Descent, and Gradient Descent work to adjust model parameters to minimize errors. When choosing an optimizer, several factors such as memory demands, convergence rate, and dataset size must be carefully evaluated. The key to identifying the optimal optimizer and hyperparameter combination for a specific machine learning task lies in experimenting with different options.

**INDEX**

[**ACKNOWLEDGEMENT i**](#_Toc154150777)

[**CHAPTER I: Learn and compare Optimizer methods in training machine learning models. 2**](#_Toc154150778)

[**I) Learn about Optimizer methods 2**](#_Toc154150779)

[**A) Gradient Descent (GD) 2**](#_Toc154150780)

[**B) Adaptive Moment Estimation (ADAM) 3**](#_Toc154150784)

[**C) Root Mean Square Propagation (RMSprop) 3**](#_Toc154150785)

[**D) Adaptive Gradient Algorithm(AdaGrad) 3**](#_Toc154150786)

[**II) Compare each Optimizer methods 4**](#_Toc154150787)

[**CHAPTER II: Learn about Continuous Learning and Test Production when building a machine learning solution to solve a problem. 5**](#_Toc154150788)

[**A) Countinuois Learning 5**](#_Toc154150789)

[**B) Test Production 5**](#_Toc154150790)

[**REFERENCES 6**](#_Toc154150791)

CHAPTER I: Learn and compare Optimizer methods in training machine learning models.

\*What is Optimizer in Machine Learning?

In the environment of technology and computer science, Optimizer is a popular term used to refer all agorithms or tool with the optimize a function or system.

In Machine Learning and Deep Learning, Optimizer is a vital component in machine learning model training process, with the primary aim idd adjust the weights and parameters of model to diminish the loss function, it means that the error prediction performance between prrdicted output and actual result.

Using optimizer in Machine Learning is extremely essential cause their benefit in helping model learn from sub data và adjust the paramenter to get the best performance.

I) Learn about Optimizer methods

A) Gradient Descent (GD)

1) Introduction

Gradient Descent is a optimial algorithm used in adjusting the parameters from the model based on derivative of loss fnction. The goal of Gradient Descent is to find the smallest value of loss function by follow the opposite direction of loss function derivative.

The formula: ****xnew = xold – learningrate.gradient(x)****

A diagram of a graph

Description automatically generatedGD in from a physical perspective

2) How does Gradient Descent work?

There are 4 steps in the the process how Gradient Descent work:

1. Initialize Parameter:

To begin with, the parameters of models will be initialized randomly or by other way was set before.

1. Caculate derivative:

Then, the gradient descent will compute the derivative of the cost function by each parameters. This derivative will give the direction and the level of change from the loss function when we change the value of parameter.

1. Revise parameters:

Accoroding to the calculated derivative, the algorithm will revise the value of each parameter by the opposite direction with derivative with learning rate in certain. This means that can reduce the loss function in minimum and leads to better parameter value.

1. Repeat process:

The process is repeated until the parameter is updated and got the stop conditions, such as the maximum iteration is defined before or the value of loss function interavtive equals 0.

3) What is Learning Rate?

In the field of Machine Learning, Learning Rate is a crutial hyperparameter in the optimal process when using the parameter such as Gradient Descnet.

Learning Rate is a snall positive constraint, which regulating the update rate of parameters in the model training. Besides that, learning rate also regulates the “jump” that the optimization algorithms use to move along the the derivative of loss function. If it is so big, the algorithms will move past the desired optimal point and not converge. However, in vice versa, if the learning rate is too small, the convergence process may become slowly and consume more training time.

A diagram of a graph

Description automatically generated

4) Advantages and Disadvangtages

|  |  |
| --- | --- |
| Advantages | As a classic and powerful optimization method, Gradient Descent is an important tool in machine learning and optimization.  The gradient descent method can converge to the global optimal point of the loss function, and can be effectively applied on large data and large parameter space. |
| Disadvantages | When using Gradient Descent, attention should be paid to some issues such as too small a learning rate which can lead to the algorithm not converging or taking a long time to converge.  For complex loss functions, Gradient Descent can get stuck at the local minimum rather than finding the global optimum. In the case of Batch Gradient Descent, calculating the derivative over the entire data can be expensive and slow on large data sets. |

A diagram of a gradient descent

Description automatically generated

The variation of Gradient Descent

A.1) Batch Gradient Descent (BGD)

1) Introduction

Batch Gradient Descent(BGD) is a basic optimization algorithm, usually used in fields of machine learning. In BGD, the parameters in model was updated based on average gradient of the entire of all dataset. It means that in each iterations, BGD calculates the gradient for all training model and then updates the parameters from the model.

2) How does Batch Gradient Descent work?

1. Compute Gradients:

BGD calculates the gradients of the cost function with each parameters of model by using all training dataset. This thing related to sum the gradients descent of each training example.

2. Update Parameters:

After calculating the average of gradient over the dataset, BDG will update the parameters of model by taking a step in the opposite direction, which was scaled by learning rate.

3. Repeat:

Repeating step 1 and 2 with the number of iterations until satisfy the convergence critrrion.

Note: It is important to notice that BGD is only update the paramenters after hanlde all dataset, which need to have in computationally intensive for big datasets.

3) Advantages and Disadvangtages

|  |  |
| --- | --- |
| Advantages | * Accurate convergence: BGD enables to converge accurately to the golab minimum of the loss function and between local minimum points. * Sable convergence: BGD often converges more stable than optimization algorithms based on the level of graduent such as SGD and Mini-Batch Gradient Descent. |
| Disadvantages | Firstly, BGD processes the entire training dataset to compute gradients for each parameter update, leading to high computational costs, especially for large datasets. This also necessitates substantial memory storage, which may not be feasible in some cases.  Moreover, BGD’s infrequent parameter updates after processing the entire dataset can lead to slow convergence, particularly with millions or billions of samples. This can impede the learning process and result in inefficiencies. Additionally, BGD is susceptible to local optima due to its reliance on the entire training set, hindering its ability to explore the complete solution space and converge to the optimal solution.  Furthermore, BGD may struggle with non-convex optimization problems, potentially converging to suboptimal solutions or becoming stuck in saddle points, reducing model performance. In addition, its inability to effectively handle dynamic training data, frequent additions of new samples, or dataset changes can limit its effectiveness |

A.2) Stochastic Gradient Descent (SGD)

1) Introduction

Similar with BGD, SGD is also another variation of Gradient Descent, instead of updating the weights after epoch, in each epoch has the N data points we will update the weight N times. Focusing on one side, SGD will reduce the speed of one epoch.

Thus, looking from another direction, SGD will converge extremely quickly after a few epochs. In addition, the formula of SGD is the same as GD, but it is performed on each data point.

2) How does Stochastic Gradient Descent work?

1.Initialization:

To begin with, initialize the parameters of model “weight” and “biases” with the random values.

2.Iteration Process:

Select the sample 🡪 Forward Pass 🡪 Calculate the loss 🡪 Backward Pass 🡪 Update Parameters

3.Converge:

This process keeps on with the time of iterations default before or until satisfy the converge criteria, such as small change in the loss or parameters.

Note: The vital notice is there are variations of SGD such as batch gradient descent and mini-batch gradient descent, they have their own balance in the efficiency and convergence characteristics.

3) Advantages and Disadvangtages

|  |  |
| --- | --- |
| Advantages | The algorithms can use to handle with the large dataset, which GD cannot, and this algorithm also was applied up to now. |
| Disadvantages | The algorithm has not solved 2 demerits of GD yet, so that, we must combine with some other algorithms such as AdaGrad, |

A.3) Mini Batch Gradient Descent

1) Introduction

Mini-Batch Gradient Descent is a modification of Gradient Descent, which divides the training dataset into small batches, that are used to compute model errors and update model coefficients.

2) How does Mini Batch Gradient Descent work?

1. Initialize Parameter:

To begin with, initialize the parameters of model “weight” and “biases” with the random values.

1. Mini-Batch Selection:

The mini-batch gradient descent divides the training data into small random batches of a fixed size instead of using the entire training dataset or a single example.

1. Iterative parameters:

Mini-Batch Sampling 🡪 Forward Pass 🡪 Loss Calculation 🡪 Backward Pass 🡪 Updating Parameter.

1. Converge:

The step 2 and 3 are repeated until get stopping criterion such as algorithm reach the time of iteration was specified before, the cost function is more stable or the gradient get enough small.

3) Advantages and Disadvangtages

|  |  |
| --- | --- |
| Advantages | * Mini-batch Gradient Descent takes advantage of parallelism in modern hardware, such as GPUs. It processes multiple data points simultaneously within each mini-batch. * This parallelism accelerates the gradient computation and parameter updates, leading to faster convergence. |
| Disadvantages | * It can be less stable and more sensitive to the choice of learning rate and mini-batch size. * Additionally, it may oscillate around the optimal point instead of reaching it exactly and require more tuning to find the best parameters and hyperparameters. |

B) Adaptive Moment Estimation (ADAM)

1) Introduction

Adaptive Moment Estimation is an algorithm to calculate the learning rate which satisfies each weight. Adam not only stores the average square of previous gradient like Adadelta but also saves the average value of model.

2) How does Adaptive Moment Estimation work?

1. Initialize Parameter:

To begin with, initialize the first and second moments of gradient equals 0, these moments are estimated, the first and the second moment of gradient respectively.

1. Process of interation:

Gradient Calculation 🡪 Update Bias-Corected First and Second Moment Estimates 🡪 Parameter Update.

1. Converge:

The process keeps on for a fixed number of iteration or until a converge criterion is met

3) Advantages and Disadvangtages

|  |  |
| --- | --- |
| Advantages | * Faster convergence :  By adjusting the learning rate during training, Adam converges much more quickly than SGD. * Easy to implement  : Only requiring first-order gradients, Adam is straightforward to implement and combine with deep neural networks. A few lines of code using Python and PyTorch are all you need. * Robust algorithm : Adam performs well across various model architectures. * Little memory requirements : Adam requires storing just the first and second moments of the gradients, keeping memory needs low. * Wide community adoption :  Adam is used extensively by deep learning practitioners and has become a default, go-to optimizer. |
| Disadvantages | The disadvantages of Adam include the need for larger and deeper models with more parameters to achieve similar model fits as other optimizers like the LM algorithm. |

C) Root Mean Square Propagation (RMSprop)

1) Introduction

A diagram of a diagram of a diagram

Description automatically generated

Root Mean Square Propagation is an optimization algorithm, was used in training nerual metworks. It was designed to hanlde some of the limitations of Adagrad algorithms by changing the way that the historical gradient infrformation is accumualted.

2) How does Root Mean Square Propagation work?

1. Initialize Parameter:

To begin with, initialize a leaky, runiing average of squared gradient G\_t.

G\_t as G\_t = βG\_{t-1} + (1-β)(∇θ J(θ\_t))^2

Where G\_0 = ) and β is a decay rate, set to 0.9

1. Process of interation:

Gradient Calculation 🡪 Adaptive Learning Rate🡪 Parameter Update.

1. Converge:

The process keeps on for a fixed number of iteration or until a converge criterion is met

3) Advantages and Disadvangtages

|  |  |
| --- | --- |
| Advantages | * RMSprop, through automatic learning rate adjustment based on gradient averages, accelerates convergence and mitigates the risk of 'exploding' or 'vanishing' gradients. Its proficiency is particularly notable when optimizing non-stationary objectives, outperforming other algorithms due to its adaptability to unstable objectives. * By minimizing the influence of abrupt gradient variations via a moving average approach, RMSprop fosters stability, curbing training fluctuations and promoting consistent progress. |
| Disadvantages | * While RMSprop offers advantages, it requires careful selection of the decay\_rate parameter. If set too high, information is forgotten too quickly, leading to instability, whereas a too low setting can slow down convergence. * Additionally, like many optimization methods, RMSprop does not assure precise convergence to the global minimum. The outcome is contingent on factors such as initial initialization, learning rate, and model structure. |

D) Adaptive Gradient Algorithm(AdaGrad)

1) Introduction

Adagrad is also an algorithm for optimization based on gradient descent, which adjusts the learning rate for different features, making smaller updates (i.e. low learning rate) to the parameters. numbers related to frequently occurring features and larger updates (i.e. high learning rate) for parameters related to infrequent features. For this reason, Adagrad is well suited for handling sparse data.

2) How does Adaptive Gradient Algorithm work?

1. Initialization:

To begin with, initializes a sum of square of gradient equal 0 for each parameter.

2. Gradient Calculation:

With each iteration in the training process, AdaGrad calculates the gradient of cost function with respect to parameters of models.

3. Accumulating Squared Gradient:

Accumulates the squares of the gradients for each parameter over time. This is done by adding the square of each gradient in historical sum of squares for the corresponding parameter.

4. Learning Rate Adjustment:

Adjusted based on the accumulated squared gradients. The learning rate for each parameter is divided by the square root of the sum of the squared gradients for that parameter up to that point in time.

5. Updating Parameter:

The model parameters are updated using the adjusted learning rates and the gradients, effectively scaling the updates based on the historical behavior of the gradients.

3) Advantages and Disadvangtages

|  |  |
| --- | --- |
| Advantages | * Determining whether the gradient is deemed sufficiently large is not necessary. * The value is automatically modified based on the gradient's magnitude. While other coordinates with smaller gradients are handled more leniently, regular coordinates with huge gradients are drastically decreased. |
| Disadvantages | * The learning rate of the Adaptive Gradient Algorithm (AdaGrad) in machine learning is continuously dropping, which is its primary drawback. * The learning rate is divided by an ever-larger number as a result of AdaGrad's gradual accumulation of the gradient's square, which dramatically lowers the model's rate of convergence. This may result in learning process delays and increase the likelihood that the model will not converge to the intended point. * This latency can greatly affect model performance, particularly in deep learning models. AdaGrad variants have been created to address this limitation, but their use has to be carefully evaluated in light of the specific data set and model properties. |

II) Compare each Optimizer methods

1. Compare in table

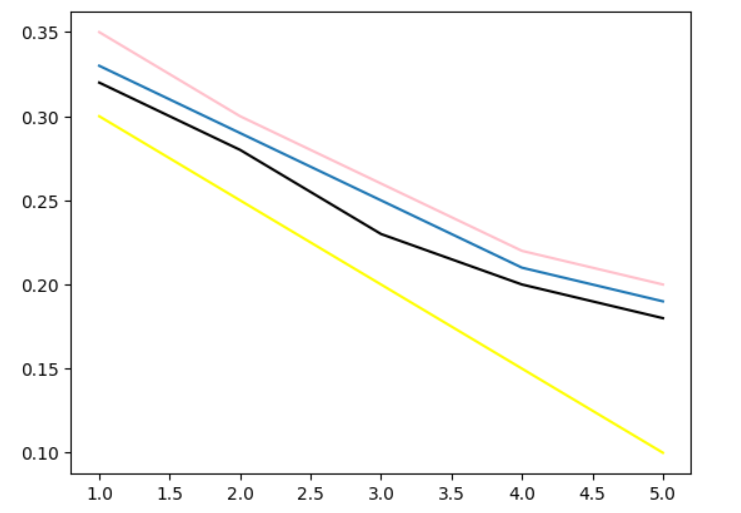
|  |  |
| --- | --- |
| Method | Efficiency |
| Gradient Descent (GD) | GD is a basic and popular optimization algorithm. It can perform well on convex problems and can be widely applied. |
| Adaptive Moment Estimation (ADAM) | By combining the benefits of the Momentum and RMSprop algorithms, ADAM provides a powerful technique for quick and reliable convergence on a wide range of problems. |
| Root Mean Square Propagation (RMSprop) | RMSprop helps control learning rates and ensures stable learning rates across many types of problems, especially in deep learning. |
| Adaptive Gradient Algorithm (AdaGrad) | AdaGrad can automatically modify the learning rate based on each parameter, making it appropriate for issues with sparse data. |

1. Compare in graphical

\*Code:

A screenshot of a computer code

Description automatically generated

\*Result:

CHAPTER II: Learn about Continuous Learning and Test Production when building a machine learning solution to solve a problem.

1. Countinuois Learning
2. Introduction

Continuous Learning, also known as lifelong learning, incremental learning or online learning is a method of machine learning, in which the model continuously learns from the new data it meets without the training from the beginning.

In continual learning, the model must adapt with the new data and maintain knowledge which is learned before.

Nowadays, continuous learning is a hot potato for researching in the field of machine learning, especially when applied to AI (artificial intelligence) systems.

A screenshot of a computer screen

Description automatically generated

**Some variation of Continual Learning**

2) How does Continuous Learning work when building a machine learning solution to solve a problem.

1. Data Stream Integration

The machine learning solution is designed to continuously ingest and process new data as it becomes available, adapting to the changing data stream. This often involves building scalable data pipelines and real-time data processing capabilities.

2. Incremental Model Training

The approach is made to include incremental learning, in which the model is updated and retrained when new data becomes available, as opposed to training it on a fixed dataset. To do this, methods for effectively updating the model parameters while retaining the knowledge acquired earlier must be developed.

3. Adaptation to Concept Drift

Continuous Learning solutions need to be able to adapt to concept drift, where the statistical properties of the data change over time. This may involve monitoring data distribution shifts and updating the model accordingly.

4. Catastrophic Forgetting Mitigation

To prevent catastrophic forgetting, techniques like regularization, rehearsal, and architectural changes are used to make sure the model retains its performance on tasks it has already learnt while adjusting to new input.

5. Dynamic Model Expansion:

The architecture of the model is intended to evolve dynamically in order to include new information in situations when the challenge involves an increasing number of classes or new kinds of data.

6. Automatic Model Evaluation and Refinement:

Continuous Learning solutions frequently incorporate processes that automatically assess and improve the model's performance over time, determining when more fine-tuning or retraining of the model is required.

7. Memory Augmentation and Context Retention:   
The model is assisted in retaining significant contextual information from historical data by using strategies like external memory augmentation and attention processes, which help with knowledge retention over time.

8. Feedback Loop Integration:

The solution incorporates feedback loops to continuously learn from its own predictions and adapt to evolving requirements or user behavior.

9. Deployment Infrastructure:

Models for continuous learning must be implemented on an architecture that can accommodate the smooth integration and updating of new models while maintaining a high level of predictability.

10. Ethical and Regulatory Considerations:

Continuous Learning solutions must adhere to ethical and regulatory standards, especially in sensitive domains where model performance degradation or unintended consequences could have significant impact.

3) Advantages and Disadvangtages

|  |  |
| --- | --- |
| Advantages | The ability to generalize and make better predictions is derived from the accumulation of knowledge over time. By retaining and building upon learned knowledge, a system can adapt well to new data and incorporate new knowledge effectively. This continuous accumulation and adaptation process enables the system to enhance its predictive abilities and maintain relevance in dynamic environments. |
| Disadvantages | Catastrophic Forgetting and Concept Drift are significant issues in continual learning, leading to performance decline and adaptation challenges. Continuous learning methods also require substantial data volumes, leading to higher costs, while managing accumulated knowledge from diverse sources demands sophisticated systems. |

**4)** **Application of Continuous Learning**

- Adaptive Robotics

- Anomaly Detection

- Medical Diagnostics

- Predictive Maintenance

-Natural Language Processing

- Personalized Recommendations

1. Test Production
2. Introduction

In the field of machine learning, Test Production always refers to deploying and evaluating the machine learning models in the environment of production. This process encompasses the utilization of a meticulously trained model that has undergone rigorous testing on historical data. Subsequently, the model is deployed to perform predictions or classifications on fresh, real-world data, thereby contributing to informed decision-making and insightful analysis in practical scenarios.

Since test production offers assessment data on model performance on fresh data, it is crucial to the development of machine learning systems.

1. **Step of the Test Production process**
2. **Collect and preprocess data:**

Collecting the related dataset to train and testing the model, or also preparing for analysis.

1. **Train model:**

Using algorithms and training data to build the model of machine learning.

1. **Evaluate model:**

Rating the performance of training model by using test data to make sure it has the best generalization or unseen data.

1. **Deploy and test:**

Implementing training-model in the real-environment or simulated-environment to estimate the performance of this model on the real-time data.

1. **Monitoring and iteration:**

Monitoring the performance of model continuously in the environment of production and updating or retraining the model when need to maintain it effectively.

REFERENCES

Vietnamese

<https://topdev.vn/blog/thuat-toan-gradient-descent/#gradient-descent-la-gi>

<https://ichi.pro/vi/gradient-descent-thiet-ke-mo-hinh-hoc-may-dau-tien-cua-ban-190503756310879>

https://tek4.vn/khoa-hoc/machine-learning-co-ban/thuat-toan-adagrad-va-van-de-hieu-chinh-learning-rate

English

<https://machinelearningmastery.com/gradient-descent-for-machine-learning/>

https://machinelearningmastery.com/gradient-descent-with-rmsprop-from-scratch/

<https://www.databricks.com/glossary/adagrad>

<https://www.sciencedirect.com/science/article/pii/S0893608019300231>

<https://wiki.continualai.org/the-continualai-wiki/introduction-to-continual-learning>

<https://stats.stackexchange.com/questions/488017/understanding-mini-batch-gradient-descent>