VIETNAM GENERAL CONFEDERATION OF LABOR

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**



**NGUYỄN TIẾN ĐẠT - 520H0527**

**FINAL REPORT**

**INTRODUCTION TO**

**MACHINE LEARNING**

# **HO CHI MINH CITY, YEAR 2023**

VIETNAM GENERAL CONFEDERATION OF LABOR

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**



**NGUYỄN TIẾN ĐẠT - 520H0527**

**FINAL REPORT**

**INTRODUCTION TO**

**MACHINE LEARNING**

*Instructor*:

**Mr. Lê Anh Cường**

**HO CHI MINH CITY, YEAR 2023**

**TABLE OF CONTENT**

[CHAPTER 1. Learn and compare Optimizer methods in training machine learning models. 2](#_Toc27736)

[1.1 Gradient Descent 2](#_Toc15320)

[1.2 Adagrad 4](#_Toc11360)

[1.4 Compare Optimizer methods 6](#_Toc6622)

[CHAPTER 2. Learn about Continuous Learning and Test Production when building a machine learning solution to solve a problem. 10](#_Toc16278)

[2.1 Continual Learning 10](#_Toc16772)

[2.2 Test Production 13](#_Toc26985)

**TABLE OF IMAGE**

[Image 1 - Gradient Descent 4](#_Toc18604)

[Image 2 - Batch Gradient Descent 9](#_Toc15709)

[Image 3 - Stochastic Gradient Descent 10](#_Toc32309)

[Image 4 - AdaGrad 10](#_Toc20470)

[Image 5 - Adam 11](#_Toc11700)

# Learn and compare Optimizer methods in training machine learning models.

## Gradient Descent

Gradient descent is a basic optimization technique used in machine learning and optimization. It is help to gradually update the model weights depending on the derivative of the loss function in order to minimizes the loss function and reach the optimal point (Local minima) or better, a Global Minima.

The formula for updating the weight of Gradient Descent:

****

: are the weights at time t + 1.

: are the weights at time t.

: is the learning rate.

: is the gradient of the loss function at .

Gradient descent has several variations based on the amount of data used to calculate the gradient of the loss function including Batch Gradient Descent, Stochastic Gradient Descent and Mini-batch Gradient Descent.



Image 1 - Gradient Descent

### Batch Gradient Descent

Batch Gradient Descent (BGD) is an optimization method that updates weights based on the entire training data set. Before updating the weight set, all data points are used to calculate the gradient.

The formula for updating the weight of Batch Gradient Descent:

****

: are the weights at time t + 1.

: are the weights at time t.

: is the learning rate.

: is the gradient of the loss function at in the dataset X.

### Stochastic Gradient Descent

Stochastic Gradient Descent (SGD) updates weights based on each data sample and corresponding label . For each data sample in the training set, SGD computes the gradient of the loss function, and in each epoch with N data samples we will update the weight N times.

The formula for updating the weight of Stochastic Gradient Descent:

****

: are the weights at time t + 1.

: are the weights at time t.

: is the learning rate.

: is the gradient of the loss function at in the data sample with label .

### Mini-batch Gradient Descent

Mini-batch Gradient Descent (MBGD) will update based on each small batch (mini-batch) of data. Mini-batch GD updates the weight set using k data points

(1 < k < N, where N is the total number of data points).

The formula for updating the weight of Mini-batch Gradient Descent:

****

: are the weights at time t + 1.

: are the weights at time t.

: is the learning rate.

: is the gradient of the loss function at in the k data points.

## Adagrad

Adagrad is a gradient optimization method used during the training of machine learning models. It is intended to automatically adjust the learning rate of each parameter based on the gradient's history.

Adagrad's learning rate varies according to the weights: a low rate for weights corresponding to common features and a high rate for weights corresponding to rare features.

The formula for updating the weight of Adagrad:

****

: are the weights at time t + 1.

: are the weights at time t.

: is the initial learning rate.

: is the current gradient of the loss function.

: is the sum of squares of the gradient.

: is a small number (usually added to avoid division by zero).

## Adaptive Moment Estimation (Adam)

Adam (Adaptive Moment Estimation) is an algorithm that calculates the adaptive learning rate for each weight.

Instead of using the entire data set to calculate the gradient, this optimization algorithm creates a stochastic approximation using a randomly selected data subset.

Adam stores the average square of the previous gradients and also stores the average momentum value.

The formula for updating the weight of Adam:

****

: are the weights at time t + 1.

: are the weights at time t.

: is the learning rate.

: is a small number (usually added to avoid division by zero).

: is the adjusted nominal value of the momentum coefficient.

: is the nominal value of the momentum coefficient of the squared gradient

The formula for and :

**=**

**=**

: is the momentum coefficient calculated from the gradient.

: is the square of the gradient.

: is the number of iterations (epoch).

& : are the momentum adjustment parameters to reduce momentum and reduce fluctuations.

## Compare Optimizer methods

|  |  |  |
| --- | --- | --- |
| **Methods** | **Advantages** | **Disavantages** |
| Batch Gradient Descent (BGD) | Ensure convergence to the global optimal point. | Computing on the entire data, not suitable for large data, requires large memory.  When the data set is large, calculating the gradient takes a long time and incurs a high computational cost. |
| Stochastic Gradient Descent  (SGD) | Effective with large data, suitable for problems that need to be learned quickly. | Convergence to the global minima point is not guaranteed, it fluctuates a lot. |
| Mini-batch Gradient Descent | Combining the advantages of BGD and SGD, effective with big data. | It is necessary to set the mini-batch size, which does not guarantee global minima convergence. |
| Adagrad | Optimize the learning rate for each parameter separately, suitable for parameters that require different learning rates. | The learning rate decreases too quickly, which can lead to getting stuck at the local minima.  Over time, the sum of squared variations will grow larger until it causes the learning rate to become extremely low, causing training to stop entirely. |
| Adaptive Moment Estimation | Combining both momentums and adaptive learning rate, popular and effective in many cases. | Hyperparameters may need to be tuned, which sometimes does not work well with all data types. |

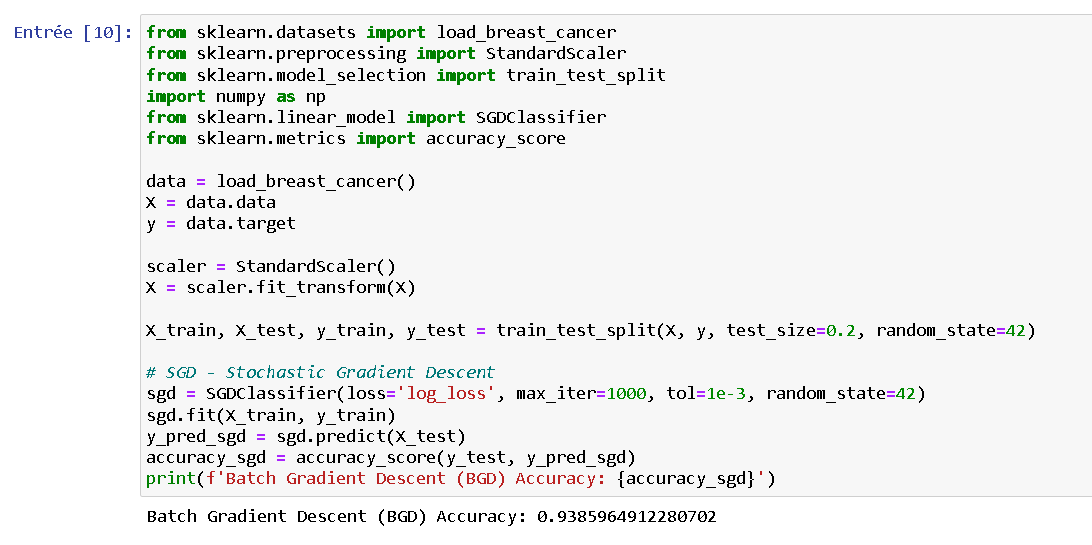


Image 2 - Batch Gradient Descent

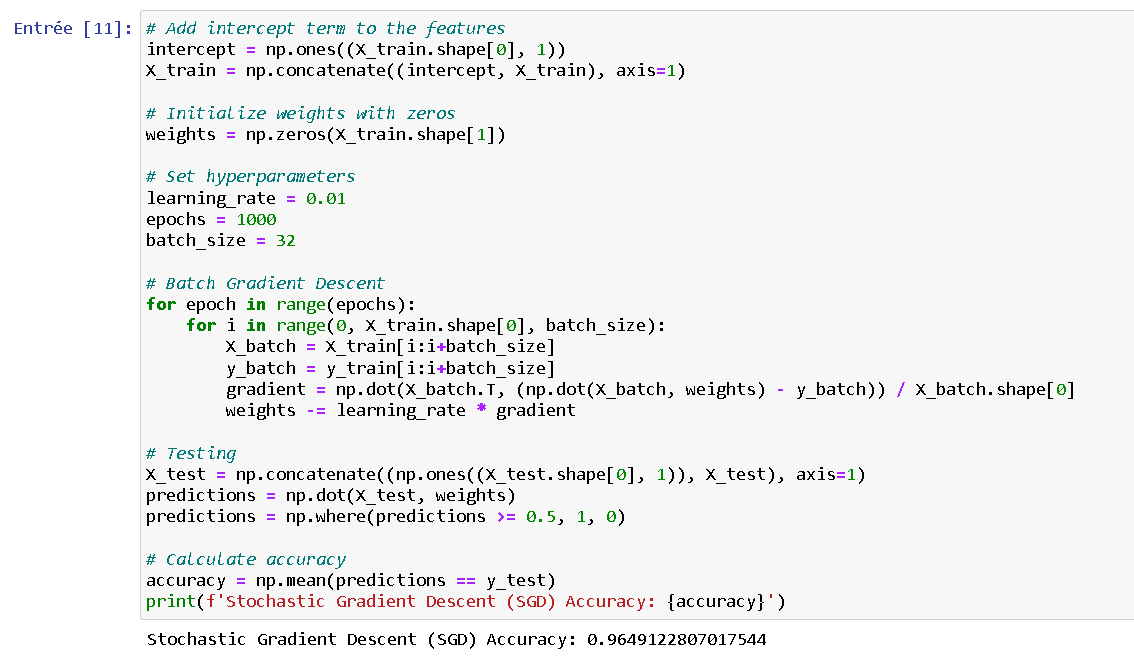


Image 3 - Stochastic Gradient Descent

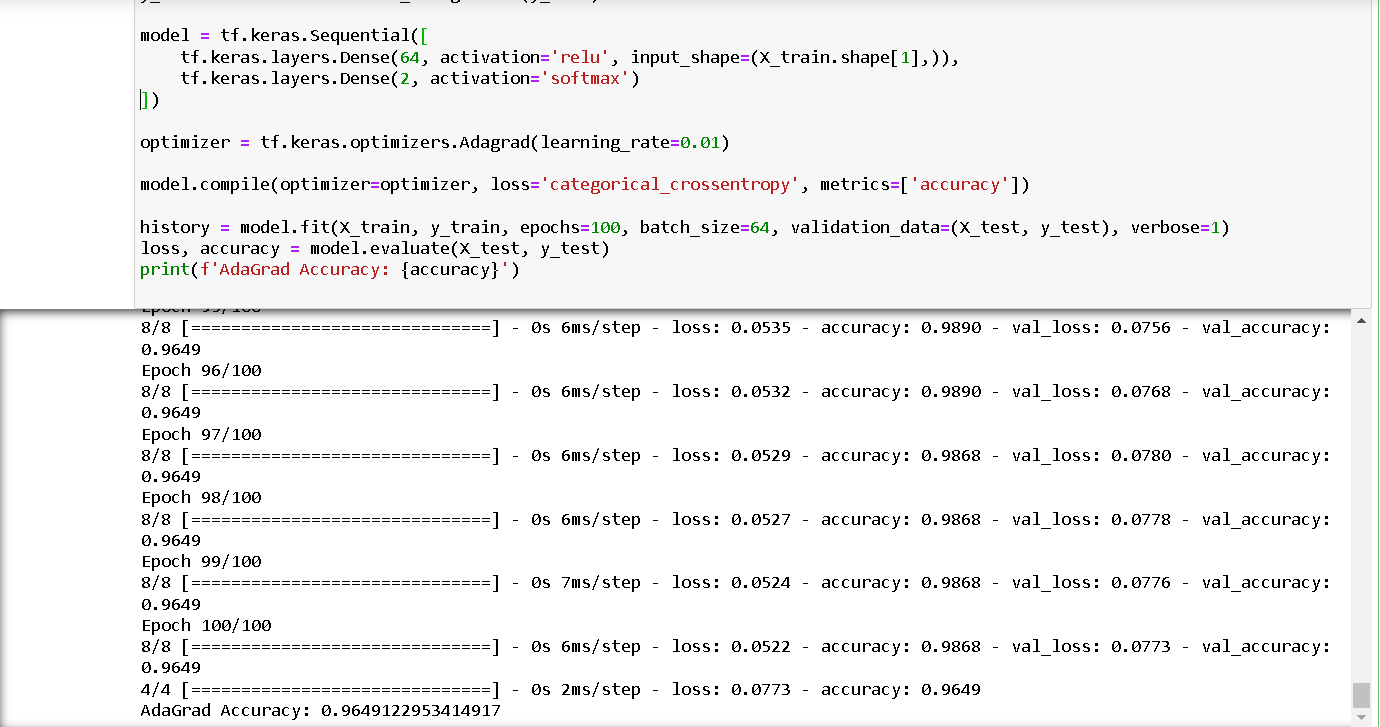


Image 4 - AdaGrad

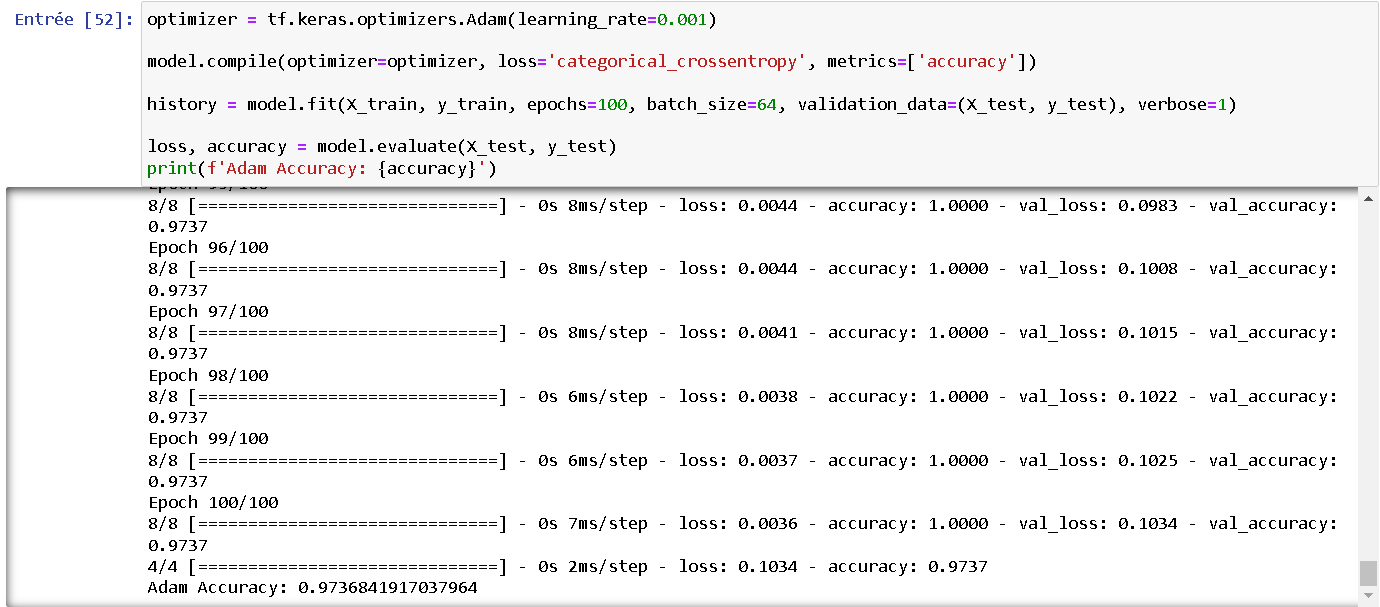


Image 5 - Adam

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

## Continual Learning

Continual Learning is a field of machine learning research that focuses on the ability of computer systems or machine learning models to continuously learn from new data while forgetting old knowledge.

Continual Learning seeks to create models that can continuously learn from available data and expand its knowledge when exposed to new information, in contrast to traditional machine learning, which frequently requires specific training data for each specific task.

### Challenge

#### Lapse Forgetting

When a machine learning system model is exposed to new data, it can lose knowledge it has learned. This frequently occurs when the model is confronted with new tasks or data in the absence of special mechanisms or methods for retaining old information.

This often occurs due to competition between new information and old information in the model's memory or learning space. As new information is introduced, part of the model's learning space may be updated to reflect the new data, and this may cause old information to be pushed out of the model's memory, leading to to the phenomenon of contagious forgetting.

#### Interference

Interference can occur when information from new data is similar or contradicts previously learned information. When a continual learning model has to process new data, its weights can be adjusted to reflect this new information. However, this can result in the model being unable to distinguish between new and old information.

It can degrade the model's ability to leverage old knowledge to solve new tasks, especially when new data overlaps or conflicts with learned data.

#### Scalability

In Continual Learning, scalability refers to a machine learning model's ability to maintain and improve performance in the face of large and diverse amounts of data.

This includes the ability to handle data from a variety of sources, in large quantities, and with a high level of complexity without sacrificing model performance or quality.

Some factors to consider in evaluating the scalability of a continuous learning system include: Efficiency, Flexibility and Resource management.

### Improvement Methods

#### Experience Replay:

The mechanism of Experience Replay is to store learning instances from the past into a memory and then use them for reuse when the model is learning from new data. Instead of just learning from the latest data, the model will sometimes retrieve instances from the past to learn from them.

This helps the model learn from old instances it may have forgotten, minimizing contagious forgetting.

#### Regularization based on Memory:

Memory-based Regularization stores important weights from previous tasks into memory and then uses this information to minimize over-adjusting these weights as the model learns from new data.

As a result, the continuous learning model can retain and prioritize the maintenance of critical knowledge gained from previous tasks.

Regularization based on Memory emphasizes retaining important knowledge from previous tasks and minimizing forgetting of this important information when the model is presented with new data.

#### External Memory Architectures

External Memory Architectures is an approach that focuses on using external memory to model and store important information from previous tasks.

This approach allows the model to access and update information from external memory independently of the learning process in the main model.

External Memory Architectures create a flexible environment for a machine learning model, allowing it to store and use information from previous tasks to improve performance and generalization when learning from new data.

## Test Production

Test Production is an important part of the Machine Learning model deployment process. It ensures that the trained Machine Learning model can perform effectively when deployed into a real production environment.

It also helps identify problems that may occur during implementation and provide corrective measures before the model is widely deployed.

### Test Production implementation process

Data Preparation: Be sure to collect data from sources that are reliable and large enough to reflect every aspect of the production environment. After that, check, clean, and normalize data to ensure it is high quality and suitable for testing.

Prepare Test Environment: Set up an environment that is as similar to the production environment as possible. This includes configuring the system, installing the software environment, and creating conditions similar to the actual environment. After that, applying the trained Machine Learning model into the prepared test environment to start the testing process.

Testing and Evaluation: Perform a variety of tests to ensure the model performs as expected. This includes accuracy testing, response time evaluation, speed testing, and other evaluation metrics depending on the model's application context. Then, measuring model performance in a test environment and compare against predetermined criteria.

Handling Issues: If the model does not perform as expected during testing, it is necessary to adjust the model to improve performance. If errors are detected in the system or deployment process, repairs and adjustments should be made to ensure the stability and reliability of the model.

Monitoring and Updates: If the model has passed the tests and meets the requirements, it will move into the production environment for real-world use. Then, monitoring model performance and make updates or adjustments as necessary to maintain and improve performance.

### Some types of Production Testing

End-to-End Testing: Checking the entire process of a Machine Learning model from data processing, model training to predicting results to ensure that the model performs as expected in a production environment. Test all stages of the Machine Learning process to ensure completeness and accuracy.

Performance Testing: Measure the performance of Machine Learning models in real-world environments by evaluating performance metrics such as response time, processing speed, and scalability to ensure the model can meet application requirements in real-world conditions.

A/B Testing: Comparing the performance of the new Machine Learning model with another version or approach in a real-world environment by comparing variations of the model or method to determine whether the new approach improves performance. Through comparison and evaluation, decide whether the new model should be widely deployed or not.

Regression Testing: Check that deploying or updating a new model does not affect features that worked well before. If implementing a new model causes new bugs or problems, Regression Testing will help detect and resolve them promptly to ensure that previously working features still work as expected after changes in the model.