VIETNAM GENERAL CONFEDERATION OF LABOR

TON DUC THANG UNIVERSITY

FACULTY OF INFORMATION TECHNOLOGY



**FINAL REPORT**

**INTRODUCTION TO MACHINE LEARNING**

**Optimizer in machine learning**

*Instructor:***LÊ ANH CƯỜNG**

*Student:* **NGUYỄN QUỲNH NHƯ – 520H0562**

Course **: 24**

**THÀNH PHỐ HỒ CHÍ MINH, 2023**

VIETNAM GENERAL CONFEDERATION OF LABOR

TON DUC THANG UNIVERSITY

FACULTY OF INFORMATION TECHNOLOGY



**FINAL REPORT**

**INTRODUCTION TO MACHINE LEARNING**

**Optimizer in machine learning**

*Instructor:***LÊ ANH CƯỜNG**

*Student:* **NGUYỄN QUỲNH NHƯ – 520H0562**

Course **: 24**

**THÀNH PHỐ HỒ CHÍ MINH, 2023**

ACKNOWLEDGEMENTS

After a semester of studying Introduction to Machine learning at Ton Duc Thang University, we want to give our sincere thanks to our teachers and Faculty of Information Technology for bringing a desired condition for students able to complete their course with real-life problem seminars.

We want to express our gratitude for our lecturer – Lê Anh Cường - who helped us a lot on this course, giving us information about Machine learning so that we can improve our knowledge, our planning skill, and analysing skills. We have used this useful technique our others course for the final project, and happily, we got a great score.

Once again, we truly grateful for everyone that helps us. Maybe this report has some mistakes because this is our first report on this course, so we really want to take the comments of the teacher and lecturer for our improvement in future projects.

With sincere thanks.

THE PROJECT IS COMPLETED

AT TON DUC THANG UNIVERSITY

I assure this is my own project with the instruction of Lê Anh Cường. All the researches, the results in this report are trustworthy and have never been announced in any appearance before. The data in tables for analysis, comments, evaluations were collected by the student in many different sources, which have clearly written in references.

Besides, I used some comments, evaluations, analysis and data of other writer, organizations in the project – which is also in the citations and source notes.

**If there is any fraud in my project, I will take full responsibility for my report content.** Ton Duc Thang University is not related to the copyright infringement that I made during the implementation process (if available).

Ho Chi Minh City, 24th December 2023

Author

*(Full name and signature)*

**INSTRUCTOR VERIFICATION AND EVALUATION SECTION**

**Confirmation from the instructor**

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

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

**The teacher's evaluation part marks the test**

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

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

TABLE OF CONTENTS

[ACKNOWLEDGEMENTS i](#_Toc153979067)

[**INSTRUCTOR VERIFICATION AND EVALUATION SECTION** iii](#_Toc153979068)

[TABLE OF CONTENTS 1](#_Toc153979069)

[Task I: Trình bày bài nghiên cứu 2](#_Toc153979070)

[1. Optimizer Methods in Model Training 2](#_Toc153979071)

[Gradient Descent 2](#_Toc153979072)

[Stochastic Gradient Descent 3](#_Toc153979073)

[Gradient Descent with Momentum 4](#_Toc153979074)

[RMSProp Optimization (Root Mean Square Propogation) 5](#_Toc153979075)

[Adam Optimization 6](#_Toc153979076)

[Adagrad 6](#_Toc153979077)

[Adadelta 7](#_Toc153979078)

[Summary 7](#_Toc153979079)

[2. Continual Learning và Test Production 9](#_Toc153979080)

[Continual Learning 9](#_Toc153979081)

[Test Production 10](#_Toc153979082)

[Conclusion 10](#_Toc153979083)

[REFERENCES 11](#_Toc153979084)

Task I: Requirement

Yêu cầu: Trình bày một bài nghiên cứu, đánh giá về các vấn đề:

* Tìm hiểu, so sánh các phương pháp Optimizer trong huấn luyện mô hình học máy
* Tìm hiểu về Continual Learning và Test Production khi xây dựng một giải pháp học máy để giải quyết một bài toán nào đó.

## Optimizer Methods in Model Training

### Gradient Descent

Gradient descent is an optimization algorithm used to adjust the weights of a machine learning model to minimize the value of the loss function. The goal of the algorithm is to find the values of weights and biases so that the value of the loss function is minimized, enabling the problem to be solved optimally.

To find the minimum point, we use the derivative. Since the problem may involve many weights, instead of finding the exact minimum, we only find points with values close to the minimum by trying many solutions using the formula:

Where:

* : is learning rate
* Gradient(x is a multidimensional vector containing all partial derivatives with respect to the parameter x

To implement the GD method, we need to go through the following steps:

* Identify the loss function that need to be optimized
* Initialize the parameters to be optimized with initial values (assign gán w0)
* Initialize the parameter learning\_rate
* Iterate through the following calculation steps:
  + Calculate the derivative of the function at the current point (x).
  + Update the parameters in the opposite direction of the derivative to move left or right to get closer to the minimum.
  + Repeat the process until the stopping condition is met or the necessary number of iterations is reached.

### Stochastic Gradient Descent

Stochastic Gradient Descent (SGD) is a variation of GD, where instead of updating the weight parameters once for each epoch, the gradient will be calculated for each data point and the weights are updated N times, with N being the number of data points.

Due to the nature of updating the weights multiple times, the runtime of each epoch in SGD will be longer than with GD. However in return, the number of epochs required to achieve good results is much lower than GD.

The steps to implement the SGD method are as follows:

* Identify the loss function that need to be optimized
* Initialize the parameters to be optimized with initial values (assign gán w0)
* Initialize the parameter learning\_rate
* Iterate through the following calculation steps:
  + Shuffle the train\_data to make randomness during training
  + For each data point in the set, perform the following steps:
    - Calculate the gradient of the loss function with respect to the parameters using the batch
    - Update the model parameters in the negative direction (-)
  + Evaluate the difference in the loss function between iterations
  + If the evaluation meets the stopping condition or the number of iterations is sufficient, then stop the loop.

### Gradient Descent with Momentum

Momentum is also a variation of Gradient Descent designed to address the problem when the returned result is the local minimum rather than the minimum of the function. It is one of the most popular optimization algorithms, widely used in many modern models. To achieve this, each time the 'steps' are updated, a value like inertia is added to prevent mistaking local minima. The formula is as follows:

Where:

* Vj: The gradient retained from previous iteration
* α is the percentage rate of the retained gradient at each iteration
* L: loss function
* η: learning rate.

Steps to implement GD with Momentum:

* X Identify the loss function that need to be optimized
* Initialize the parameters to be optimized with initial values (assign gán w0)
* Initialize the parameter learning\_rate
* Iterate through the following calculation steps:
* Initialize the initial velocity value gama (usually is 0)
* Iterate through the following calculation steps:
  + Calculate the gradient of the function at the current point x.
  + Update the momentum variable by multiplying the gradient by a velocity coefficient.
  + Update the model parameters using the accumulated momentum value and adding the learning rate multiplied by the gradient.
  + Repeat the process until the stopping condition is met or the necessary number of iterations is reached.

The final result is the optimized parameters of the model.

### RMSProp Optimization (Root Mean Square Propogation)

RMSProp uses the root mean square of the gradient to normalize. This helps balance the magnitude of the step (momentum) - reducing the step for large gradients to avoid the Exploding Gradient phenomenon and increasing the step for small gradients to avoid the Vanishing Gradient phenomenon. RMSProp automatically adjusts the learning rate and selects a different learning rate for each parameter.

The method for updating the weights can be expressed by the following formula:

Where:

* + 𝑠𝑡 : Accumulated square of the gradients in the past
  + 𝜌: descent paramenter
  + 𝛥𝑥𝑡 : points that changed in the model
  + 𝑔𝑡 : gradient of parameters in current iterraction
  + ϵ: A value to ensure the result is not divided by zero.

### Adam Optimization

Adam can be considered as a combination of Momentum and RMSprop. This method calculates the learning rate for each parameter separately, using estimates of the first and second moments of the gradient to adjust the learning rate for each weight in the model. The formula for this algorithm can be written as follows:

Where:

* + vt: Moving average of the squared gradient
  + mt: Moving average of the gradient
  + β1và β2: Moving average rates.

### Adagrad

Adagrad is a method that helps decrease the learning rate by adjusting it for each parameter in a neural network. This method is an improvement by precisely adapting the learning rate based on the gradient history of each parameter, rather than using a single learning rate for all nodes.

The formula for this method is as follows::

Where:

* + Gt: is the sum of the squared gradients for iteration t
  + gt: is the gradient of the loss function with respect to parameter i at interation t
  + η is learning rate

### Adadelta

Adadelta can be seen as a variation of AdaGrad. This method does not have a learning rate parameter; instead, it uses the rate of change of the parameters themselves to adjust the learning rate. It constrains the accumulated gradient over time to a fixed-size window of the weight w.

The formula for this method is:

Where:

* + 𝑠𝑡: stores the exponentially decaying average of the second moment of the gradient
  + Δ𝑥𝑡: stores the exponentially descent average of the second moment of the parameter updates in the model
  + 𝑔’𝑡 : the square root of the ratio of the exponentially descent average of the square of the rate of change and the exponentially decaying average of the square of the gradient.

### Summary

|  |  |  |
| --- | --- | --- |
| **Method** | **Advantages** | **Disadvantages** |
| Gradient Descent | Simple, easy to understand, and solves optimization problems | Depends heavily on the initial values and learning rate |
| Stochastic Gradient Descent | Solves problems with large datasets | The runtime for each epoch is much longer. Still depends heavily on the initial values and learning rate |
| Momentum | Solve the problem of misidentified local minimum | When approaching the result, still requires several more epochs to oscillate around the target for accurate results |
| RMSProp Optimization | Solve the problem of learning speed decreasing over time, avoiding freezing | May yield only local minimum solutions instead of achieving global minimum |
| Adam Optimization | Solves the local minimum problem of RMSProp and the oscillation around the target of Momentum | Convergence is slower compared to other methods, prone to overfitting if the learning rate is not chosen correctly. |
| Adagrad | Learning rate is updated automatically | The squared parameter is added for each iteration, always positive, causing the learning rate to continuously decrease and may become arbitrarily small. |
| Adadelta | Does not require a global learning rate | Complex, involves more computations for each epoch |

**Conclusion:**

While each machine learning optimization method has its own advantages and limitations, the choice depends on the specific characteristics of the given problem and training environment.

Demo: Folder [520H0562\_NguyenQuynhNhu]

<https://github.com/giabao1912/Final_Machine_Learning>

## Continual Learning và Test Production

### Continual Learning

Continual learning, also referred to as incremental learning or life-long learning, is the idea of learning a model for a number of tasks in succession without losing the knowledge gained from earlier tasks, even in situations where the data from earlier tasks is no longer available for training new ones.

There are three scenarios for continual learning:

1. Models are always informed about which task needs to be performed. This is the easiest continual learning scenario, and we refer to it as task-incremental learning (Task-IL)
2. Models only need to solve the task in hand, they are not required to infer which task it is, then we refer it as domain-incremental learning (Domain-IL), task identity is not available at test time.
3. Models must be able to both solve each task seen so far and infer which task they are presented with. We refer to this scenario as class-incremental learning (Class-IL)

Benefits:

* Generalization: In the face of fresh data, the model becomes more resilient and accurate through continuous learning
* Retention of information: Through the use of a continuous learning approach, the model is able to gather information over time by taking into account prior knowledge acquired in earlier iterations.
* Adaptability: In the long run, a model that uses continuous learning is more predictive because it can adjust to new information, such as concept drift and emerging trends.

### Test Production

The process of creating and implementing a test to assess a machine learning model's performance on fresh data is known as test production. Making tests is essential when developing a machine learning solution to make sure the model learns from the training data and applies itself to new data effectively. The test must be created to evaluate the model's predictive abilities as well as its adaptability to new circumstances.

Benefits of Test Production in building a machine learning solution:

* Performance Evaluation and Generalization: Testing assesses the model's performance on new data, ensuring both accuracy and generalization ability.
* Overfitting Detection and Model Tuning: By comparing performance on training and test sets, overfitting can be detected, leading to necessary model adjustments.
* Confidence in Deployment and Reliability: Test results instill confidence in deploying the model, checking its reliability in real-world scenarios.
* Adaptability Testing and Continuous Improvement: Testing confirms the model's adaptability and supports continuous improvement to maintain performance and practicality.

### Conclusion

Combining Continual Learning and Test Production is essential to constructing and maintaining a flexible and effective machine learning system that can tackle diverse and continuously changing problems.

REFERENCES

[1] What is Continuous Machine Learning? (Hanna Kleinings) asscess on 20th December 2023 at <https://levity.ai/blog/what-is-continuous-machine-learning>

[2] Three scenarios for continual learning (Gido M. van de Ven, Andreas S. Tolias) access on 20th December 2024 at <https://arxiv.org/abs/1904.07734>

[3]