**VIETNAM GENERAL CONFEDERATION OF LABOR**

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**

Logo

Description automatically generated

**FINAL REPORT**

**MACHINE LEARNING**

*Instructor: Mr. Le Anh Cuong*

*Author: Dang Cao Dat – 521C0047*

*Class:*21K50301

**HO CHI MINH CITY, 2023**

# Acknowledgement

For my report of establishment, I would love to express my gratefulness and reverence to my teacher: Le Anh Cuong, who is also my mentor and the topic professor, for instructing me as well as other students to a higher level of knowledge about Machine Learning.

I also want to express my appreciation to Ton Duc Thang University's Board of Directors for providing me such favorable facilities as well as modern libraries system to complete my research of the project more easily.

Last thing is to my self-evaluation, my inexperience about knowledge is a problem of which I have to improve every day. That’s also the reason why my project is not completely perfect. Therefore, I’m ready for all your criticisms, suggestions, corrections as well as your feedback.

# DECLARATION OF AUTHORSHIP

I hereby declare that this report was carried out by myself under the guidance and supervision of Mr. Le Anh Cuong, and that the work contained and the results in it are true and have not been submitted anywhere for any previous purposes. The data and figures presented in this report 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 form other authors, and organizations have used have been acknowledged, and explicitly cited.

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

Ho Chi Minh City, Date

Author

(signature and full name)

# TEACHER’S ASSESSMENT AND CONFIRMATION

Instructor’s confirmation:

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

Ho Chi Minh City, December 19, 2023

(signature and full name)

Examiner’s confirmation:

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

Ho Chi Minh City, December 19, 2023

(signature and full name)

Table of Contents

[Acknowledgement 2](#_Toc154265915)

[DECLARATION OF AUTHORSHIP 3](#_Toc154265916)

[TEACHER’S ASSESSMENT AND CONFIRMATION 4](#_Toc154265917)

[I Model’s optimizer understanding and comparison 6](#_Toc154265918)

[I.1. Definition 6](#_Toc154265919)

[I.2. Optimizers 6](#_Toc154265920)

[I.2.1. Stochastic Gradient Descent (SGD) 6](#_Toc154265921)

[I.2.2. Momentum 6](#_Toc154265922)

[I.2.3. Adagrad 7](#_Toc154265923)

[I.2.4. RMSprop 7](#_Toc154265924)

[I.2.5. Adam 7](#_Toc154265925)

[I.3. Comparison 7](#_Toc154265926)

[II Continual Learning and Test Production 8](#_Toc154265927)

[II.1. Continual learning 8](#_Toc154265928)

[II.2. Test production 9](#_Toc154265929)

# Model’s optimizer understanding and comparison

In this chapter, we’re going to be looking at many optimizers of a training model. In real world scenarios, a model training can be accurate in some datasets, and inaccurate or having low accuracy on others. That is why some models are evaluated using different optimizers in order to help them perform better at learning a dataset’s pattern. Today, I will give you the definition and comparison between various optimizers of Feed Forward Neural Network machine learning model.

## Definition

Optimizers are algorithms that helps to build a neural network model in order to learn the features from the dataset, from which can be used to find the appropriate weight and bias to optimize the model.

## Optimizers

Now, let’s take a look at some popular optimizers:

### Stochastic Gradient Descent (SGD)

SGD is an iterative method for optimizing an objective function with suitable smoothness properties (e.g., differentiable or sub differentiable). It can be regarded as a stochastic approximation of gradient descent optimization, since it replaces the actual gradient (calculated from the entire data set) by an estimate thereof (calculated from a randomly selected subset of the data). Especially in high-dimensional optimization problems this reduces the very high computational burden, achieving faster iterations in exchange for a lower convergence rate.

### Momentum

Momentum is an extension to the gradient descent optimization algorithm that builds inertia in a search direction to overcome local minima and oscillation of noisy gradients. The main idea behind momentum is to compute an exponentially weighted average of the gradients and use that to update the weights. By taking past gradients into account, the steps to gradient descent become smoothed out, which can reduce the amount of oscillations seen in iterations.

### Adagrad

Adagrad, short for Adaptive Gradient Algorithm, is a gradient-based optimization algorithm designed to adapt the learning rate for each parameter during the optimization process, based on the past gradients observed for that parameter. The weight updating formula for Adagrad looks like: theta = theta - (alpha / sqrt(Gt + epsilon)) \* g\_t, where alpha denotes different learning rates for each weight at each iteration, Gt is the sum of the squares of the past gradients, epsilon is a small positive value number to avoid divide by zero error if in case alpha becomes 0, and g\_t is the gradient at time step t.

### RMSprop

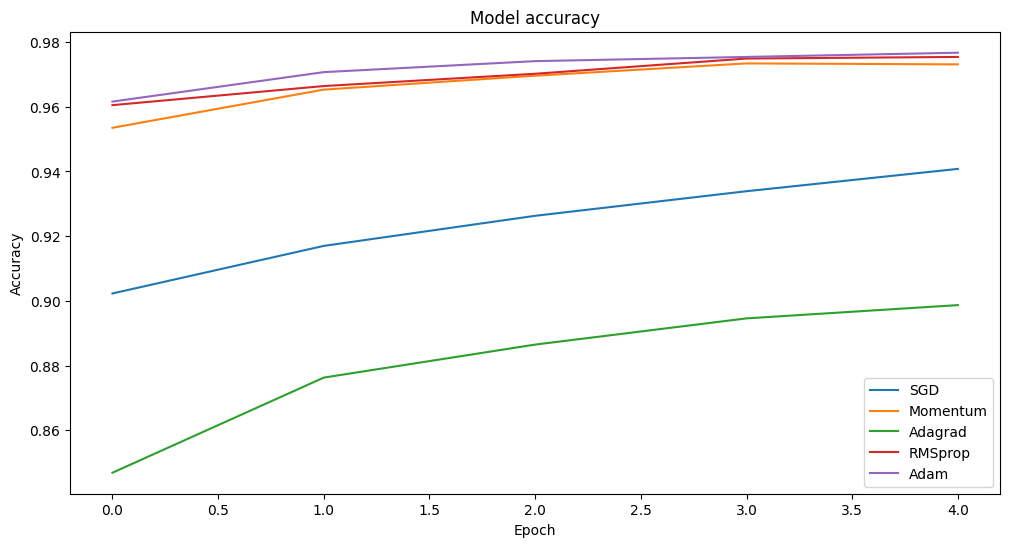
RMSprop, short for Root Mean Square Propagation, is an optimization algorithm/method designed for Artificial Neural Network (ANN) training. It is an adaptive learning rate method that uses the magnitude of the recent gradient descents to normalize the gradient. RMSProp lies in the realm of adaptive learning rate methods, which have been growing in popularity in recent years because it is the extension of Stochastic Gradient Descent (SGD) algorithm, momentum method, and the foundation of Adam algorithm.

### Adam

The Adam optimizer, short for “Adaptive Moment Estimation,” is an iterative optimization algorithm used to minimize the loss function during the training of neural networks. Adam can be looked at as a combination of RMSprop and Stochastic Gradient Descent with momentum. Developed by Diederik P.Kingma from OpenAI and Jimmy Ba from the University of Toronto in their 2015 ICLR paper (poster) titled “Adam: A Method for Stochastic Optimization“. Adam adjusts learning rates individually for each parameter, allowing for efficient optimization and convergence, especially in complex loss landscapes.

## Comparison

For the accuracy comparison between each optimizers, I use MNIST dataset. Below is the graph show the accuracy of each models through each epoch when being trained using MNIST dataset.



As it can be seen from the above graph, the accuracy of Adam, SGD(Momentum) and RMSprop are really high, around 95% to 98%, while other models, have a decent accuracy, which is now low, but not high either, with SGD from around 90% to 93%, and Adagrad, which is also the worst, has an accuracy of below 86% climbing up to nearly 89%.

# Continual Learning and Test Production

## Continual learning

Continual Learning is a process which a model learns from new data continuously without forgetting knowledge obtained from old dataset or being retrained. By updating the parameters in order to reflect the new distribution of the data, continuous learning differs completely from the traditional learning method, where models are mainly trained on a static dataset, deployed to use, and periodically retrained, making it an upgrading strategy for scientists to use in their model.

There are multiple approaches for this learning method, including incremental learning, transfer learning and lifelong learning.

The continual learning is an evolution of traditional learning methods, that means it contains all of the basic modeling techniques: pre-processing, model selection, hyper parameter tuning, training, deployment and monitoring. Besides the original steps, there are 2 additional ones which is data rehearsal and continuous learning implementation. With the help of these two steps, the model is guaranteed to learn from streams of new data efficiently without omitting the old ones.

There are many advantages and limitations for this methods. There are three most conceivable pros for continuous learning, ranging from generalization, since the model are empowered to be more robust and accurate; retention of information, as the model keep the previous knowledge from old data, allowing it to accumulate information overtime; and lastly, adaptability, because model can adapt to new data using this strategy. However, there are two main cons to limit the perfection, starting from model management, where model’s parameters are updated based on new data, resulting in new model being formed each time, making it complicated for identifying the best-performing ones. Moreover, data drift is also a concerning problem, as we have to process a large volume of new data to make the continuous learning worthwhile. That said, if the feature distribution changes abruptly, the model will risk the chance of losing predictive capabilities.

In summary, the continuous learning may be an evolutional approach for machine learning models. Even though the limitations seemed concerning, the benefits surely outweigh the disadvantages, making it the most effective way to train a model and use it in a real world scenario.

## Test production

**Test Production** in machine learning refers to the process of evaluating the performance of a machine learning model in a production environment. This involves testing the model on unseen data (also known as the test set) to assess how well the model generalizes beyond the training data. It’s crucial to ensure that the model performs well not just on the training data, but also on new, unseen data. This helps in maintaining the robustness of the model when it encounters real-world data.

There are many kinds of test production, which includes:

1. **Data Testing**: After training a machine learning algorithm using a suitable dataset, it’s crucial to evaluate its performance by exposing it to previously unseen test data. Data testing involves comparing the model’s output against actual results (also known as ground truth) for each example within the test set.
2. **Cross-validation**: This involves splitting your entire dataset into multiple smaller subsets (or folds) and iteratively training and testing your model on each fold. This ensures the model’s performance remains consistent across different portions of the data.
3. **Metrics**: Various metrics, such as accuracy, precision, recall, F1 score, or area under the receiver operating characteristic curve (AUC-ROC), can be used to measure your ML model’s performance. Each metric has its strengths and weaknesses, depending on the problem domain and dataset characteristics.
4. **Data Splitting**: A critical aspect of preparing datasets for machine learning projects is deciding how to divide them into training and testing sets. A common rule of thumb is using a 70/30 or 80/20 ratio between training data vs. testing data. However, this may vary based on factors like dataset size or complexity.
5. **Model Robustness**: You also need to perform thorough testing of your models to ensure they are robust enough for real-world encounters. This is one of the most challenging aspects of putting machine learning models into production.

Since the goal of machine learning is to create models that can generalize well to new, unseen data. So, a model that performs extremely well on training data but poorly on validation data is not a good model. Therefore, it’s important to monitor both training and validation metrics during the training process to ensure the model is learning effectively.