VIETNAM GENERAL CONFEDERATION OF LABOUR
TON DUC THANG UNIVERSITY
FACULTY OF INFORMATION TECHNOLOGY

MACHINE LEARNING

FINAL PROJECT

Supervisor: **Ph.D LE ANH CUONG**

Author: **TRAN HOANG PHUC– 521H0288**

Class : **21H50301**

Class of : **25**

HO CHI MINH CITY, 2023

VIETNAM GENERAL CONFEDERATION OF LABOUR
TON DUC THANG UNIVERSITY
FACULTY OF INFORMATION TECHNOLOGY

MACHINE LEARNING

FINAL PROJECT

Supervisor: **Ph.D LE ANH CUONG**

Author: **TRAN HOANG PHUC– 521H0288**

Class : **21H50301**

Class of : **25**

HO CHI MINH CITY, 2023

ACKNOWLEDGEMENT

To complete this final project, we would like to express our gratitude to Ton Duc Thang University for providing the necessary facilities and to the teachers who have supported us throughout our studies at the university. Moreover, we love to give thanks to our parents who gave us chances to approach knowledge at TDTU and motivated, supported us a lot.

Especially, we would like to thank Ph.D Le Anh Cuongfor teaching us with great dedication and detail, so that we have enough knowledge to use for this essay. Due to our limited experience and knowledge, we are sure that there are some mistakes in our work. We sincerely hope to receive feedbacks and constructive criticism from our teacher so that we can complete this essay more effectively.

We would like to express my heartfelt thanks and wish teacher good health.

THE PROJECT IS COMPLETED AT TON DUC THANG UNIVERSITY

I hereby declare that this is the product of our project and is guided by Le Anh Cuong. The research content 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 were collected by the author from different sources and clearly stated in the reference section. In addition, the project also uses a number of comments, assessments as well as data from other authors and other organizations, all with citations and source notes.

If any fraud is discovered, I will take full responsibility for the content of my project. Ton Duc Thang University is not involved in copyright violations caused by me during the implementation process (if any).

Ho Chi Minh City, 16th December 2033

Authors

(Signatures and full names)

Tran Hoang Phuc

SUPERVISOR’S CONFIRMATION AND EVALUATION

Supervisor’s confirmation

Ho Chi Minh City, December 16, 2023
(signature and full name)

Supervisor’s evaluation

Ho Chi Minh City, December 16, 2023
(signature and full name)

SUMMARY

In the realm of machine learning, the main goal of any models is to predict the the outcome based on the given datasets, and as for any prediction, we want the result to have as less error as possible. To combat this in machine learning, a sets of technique called optimization are used to reduce the degree of error, improving the accuracy in many models. In this report, we are going to go through some methods such as Gradient Descent, Momentum, Adagrad, RMSprop, Adaptive Moment Estimation,…

TABLE OF CONTENTS

TABLE OF PICTURES, CHARTS AND GRAPHS

CHAPTER 1 – INTRODUCTION

In machine learning model training, optimization is a crucial step to update the model's weights in order to minimize the loss function and improve performance. Various optimizer methods have been developed to address different challenges in different problems and model architectures.

CHAPTER 2 – OPTIMIZATION ALGORITHMS

2.1 Stochastic Gradient Descent

SGD is a simple and widely used method. It updates weights by moving in the opposite direction of the gradient of the loss function.

2.2 Batch Gradient Descent

Batch gradient descent is a variation of the gradient descent algorithm that calculates the error for each example in the training dataset, but only updates the model after all training examples have been evaluated.

2.3 Mini-batch Gradient Descent

Mini-batch GD is a variation of SGD where not all training data is used to update weights each time. Instead, only a small subset (mini-batch) is randomly chosen.

Mini-batch gradient descent seeks to find a balance between the robustness of stochastic gradient descent and the efficiency of batch gradient descent.

2.4 Adam(Adaptive Moment Estimation)

Adam combines information from both the gradient and the squared gradient to adaptively adjust the learning rate and momentum decay over time. This helps in quick convergence and is suitable for various types of data.

2.5 AdaGrad

Adaptive Gradients, or AdaGrad for short, is an extension of the gradient descent optimization algorithm that allows the step size in each dimension used by the optimization algorithm to be automatically adapted based on the gradients seen for the variable (partial derivatives) seen over the course of the search.

2.6 RMSprop (Root Mean Square Propagation)

Root Mean Squared Propagation, or RMSProp, is an extension of gradient descent and the AdaGrad version of gradient descent that uses a decaying average of partial gradients in the adaptation of the step size for each parameter. The use of a decaying moving average allows the algorithm to forget early gradients and focus on the most recently observed partial gradients seen during the progress of the search, overcoming the limitation of AdaGrad

2.7 Adadelta

AdaDelta is a stochastic optimization technique that allows for per-dimension learning rate method for SGD. It is an extension of Adagrad that seeks to reduce its aggressive, monotonically decreasing learning rate. Instead of accumulating all past squared gradients, Adadelta restricts the window of accumulated past gradients to a fixed size w.

2.8 Nadam

The Nesterov-accelerated Adaptive Moment Estimation, or the Nadam, algorithm is an extension to the Adaptive Movement Estimation (Adam) optimization algorithm to add Nesterov's Accelerated Gradient (NAG) or Nesterov momentum, which is an improved type of momentum.

2.9 Comparision

Stochastic Gradient Descent (SGD):

- *Pros:*
  - Simplicity and ease of implementation.
  - Computationally less expensive for large datasets.
- *Cons:*
  - Prone to convergence issues, especially in the presence of noisy or sparse gradients.
  - Requires careful tuning of the learning rate.

Batch Gradient Descent:

- *Pros:*
  - Guaranteed convergence to the global minimum for convex loss functions due to the use of the entire dataset in each iteration.

    - Stable and smooth convergence when the loss surface is well-behaved.
- *Cons:*
    - Computationally expensive, especially for large datasets, as it processes the entire dataset in each iteration.
    - Memory-intensive, as it requires storing the entire dataset in memory.
    - May struggle with non-convex loss functions or noisy gradients.

Mini-batch Gradient Descent:

- *Pros:*
    - Strikes a balance between SGD and batch gradient descent, offering efficiency and convergence stability.
    - Well-suited for a variety of dataset sizes.
- *Cons:*
    - Requires tuning of the mini-batch size, and the choice may impact convergence.

Adam (Adaptive Moment Estimation):

- *Pros:*
    - Combines the benefits of both momentum and adaptive learning rates, making it efficient and widely used.
    - Works well with sparse gradients.
- *Cons:*
    - Sensitive to hyperparameter choices, requiring careful tuning for optimal performance.

RMSprop (Root Mean Square Propagation):

- *Pros:*

    - Addresses the diminishing learning rate issue in Adagrad, making it suitable for non-stationary problems.
    - Efficient in handling varying gradient magnitudes.
- *Cons:*
    - Hyperparameter sensitivity, especially for the decay rate.

Adagrad:

- *Pros:*
    - Adapts learning rates individually for each parameter, useful for sparse data.
    - Effective in problems where some features have rare occurrences.
- *Cons:*
    - Cumulative squared gradients may lead to diminishing learning rates.
    - Not suitable for non-convex optimization problems.

Adadelta:

- *Pros:*
    - Addresses the diminishing learning rate issue in Adagrad.
    - Does not require an initial learning rate.
- *Cons:*
    - Hyperparameter sensitivity, particularly with respect to the decay rate.
    - Less commonly used compared to other optimizers.

Nadam:

- *Pros:*
    - Fast convergence due to combining adaptive learning rates from Adam and Nesterov momentum.
    - Suitable for a wide range of problems.

- *Cons:*
    - Hyperparameter sensitivity, especially the choice of beta parameters

CHAPTER 3 – CONTINUAL LEARNING AND TEST PRODUCTION

3.1 Continual Learning

Continual Learning, in the context of building machine learning solutions, refers to the capability of a model to adapt and improve its performance over time as it encounters new data. Unlike traditional machine learning scenarios where models are trained on fixed datasets, continual learning emphasizes the ability to learn from a continuous stream of information. This is particularly important in dynamic environments where the distribution of data may change, and new patterns or concepts may emerge.

Key Aspects of Continual Learning in Machine Learning Solutions:

Incremental Learning: Continual learning involves training a model on new data while retaining knowledge from previously learned tasks. This helps prevent the issue of catastrophic forgetting, where the model loses proficiency on previously learned tasks when exposed to new information.

Adaptive Architectures: Models designed for continual learning often incorporate adaptive architectures. These architectures allow the model to dynamically adjust its structure to accommodate new knowledge without sacrificing performance on existing tasks.

Regularization Techniques: Techniques such as elastic weight consolidation and parameter regularization are employed to mitigate catastrophic forgetting. These methods penalize large changes in the model parameters during training, preserving knowledge from previous tasks.

Rehearsal and Memory-Augmented Networks: Rehearsal involves periodically revisiting and training on past data to reinforce previously learned information. Memory-augmented networks, such as neural networks with external memory, enable the model to store and access relevant information from the past.

3.2 Test Production

Test Production is a crucial phase in the development of a machine learning solution. It involves designing and executing tests to evaluate the performance, robustness, and generalization capabilities of the model. Test Production is not a one-time activity but is integrated into the development life cycle to ensure the model's continuous effectiveness.

Key Aspects of Test Production in Machine Learning Solutions:

Validation and Evaluation Metrics: Define appropriate metrics for assessing the model's performance. This includes accuracy, precision, recall, F1 score, and other relevant measures depending on the nature of the problem.

Cross-Validation: Use cross-validation techniques to assess the model's generalization across different subsets of the data. This helps identify potential overfitting issues and ensures the model's reliability on unseen data.

Test Set Construction: Develop comprehensive test sets that cover a diverse range of scenarios and edge cases. This helps uncover potential weaknesses in the model and ensures it performs well in real-world situations.

Continuous Monitoring: Implement continuous monitoring and feedback loops to track the model's performance over time. This involves retesting the model periodically and updating it as needed to maintain its effectiveness in evolving conditions.

REFERENCES

ENGLISH

- Brownlee, J. (2021, January 12). Gentle introduction to the adam optimization algorithm for deep learning. MachineLearningMastery.com. https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/
- Brownlee, J. (2021b, October 11). *Gradient descent with RMSProp from scratch*. MachineLearningMastery.com. https://machinelearningmastery.com/gradient-descent-with-rmsprop-from-scratch/
- Brownlee, J. (2021b, October 11). *Gradient descent with Adagrad from scratch*. MachineLearningMastery.com. https://machinelearningmastery.com/gradient-descent-with-adagrad-from-scratch/
- Team, K. (n.d.). *Keras Documentation: Nadam*. https://keras.io/api/optimizers/Nadam/

VIETNAMESE