VIETNAM GENERAL CONFEDERATION OF LABOR

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**

Logo

Description automatically generated

**Final Project Machine Learning**

*Instructor:* **PGS.TS Lê Anh Cường**

*Student:* **Trương Gia Bảo**

*Student ID:* **521H0201**

*Class:* **21H50302**

**HO CHI MINH CITY, 2023**

VIETNAM GENERAL CONFEDERATION OF LABOR

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**

Logo

Description automatically generated

**Final Project Machine Learning**

*Instructor:* **PGS.TS Lê Anh Cường**

*Student:* ***Trương Gia Bảo***

*Student ID:* **521H0201**

*Class:* **21H50302**

**HO CHI MINH CITY, 2023**

# ACKNOWLEDGEMENT

I would like to express my sincere thanks to the teachers and lecturers of Ton Duc Thang University and especially Mr. Le Anh Cuong - lecturer of Machine Learnig Group 05 for kindly helping and supporting as well as answering our questions throughout the process of making this report, so that I can get the best results.

**THE PROJECT WAS COMPLETED**

**AT TON DUC THANG UNIVERSITY**

I would like to assure you that this is my own project and guided by 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 evaluations collected by the author himself from different sources are clearly stated in the references section.

In addition, the project also uses some comments, reviews as well as figures of other authors and other organizations with quotes and annotations of origin.

If any fraud is detected, I would like to take full responsibility for the content of my project. Ton Duc Thang University is not involved in copyright or copyright violations caused by me in the process (if any).

*Ho Chi Minh City, 29 November 2023*

*Author*

*(sign and write your full name)*

*Trương Gia Bảo*

**FACULTY ENDORSEMENTS AND REVIEWS**

**Phần xác nhận của GV hướng dẫn**

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

Tp. Hồ Chí Minh, ngày tháng năm

(ký và ghi họ tên)

**Phần đánh giá của GV chấm bài**

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

Tp. Hồ Chí Minh, ngày tháng năm

(ký và ghi họ tên)

# TABLE OF CONTENTS

[ACKNOWLEDGEMENT 1](#_Toc154081482)

[TABLE OF CONTENTS 1](#_Toc154081483)

[CHAPTER 1: Learn and compare Optimizer methods 2](#_Toc154081484)

[1.1 What is Optimizer 2](#_Toc154081485)

[1.2.1 Gradient Desent 2](#_Toc154081486)

[1.2.2 Stochastic Gradient Descent (SGD) 4](#_Toc154081487)

[1.2.3 Mini-Batch Gradient Descent 5](#_Toc154081488)

[1.2.4 SGD with Momentum 6](#_Toc154081489)

[1.2.5 AdaGrad(Adaptive Gradient Descent) 7](#_Toc154081490)

[1.2.6 RMS-Prop (Root Mean Square Propagation) 9](#_Toc154081491)

[1.2.7 AdaDelta 10](#_Toc154081492)

[1.2.8 Adam(Adaptive Moment Estimation) 12](#_Toc154081493)

[CHAPTER 2: CONTINUAL Learning and Test Production 14](#_Toc154081494)

[2.1 Continual learning 14](#_Toc154081495)

[2.2.1 Types of Continual Learning 15](#_Toc154081496)

[2.2.2 Process of Continual Learning 16](#_Toc154081497)

[2.2.3 Key aspects and challenges 18](#_Toc154081498)

[2.2 Test Production 19](#_Toc154081499)

[2.2.1 Key components of test production 19](#_Toc154081500)

[2.2.2 Step of the test production process in machine learning 20](#_Toc154081501)

[REFERENCE 23](#_Toc154081502)

# CHAPTER 1: Learn and compare Optimizer methods

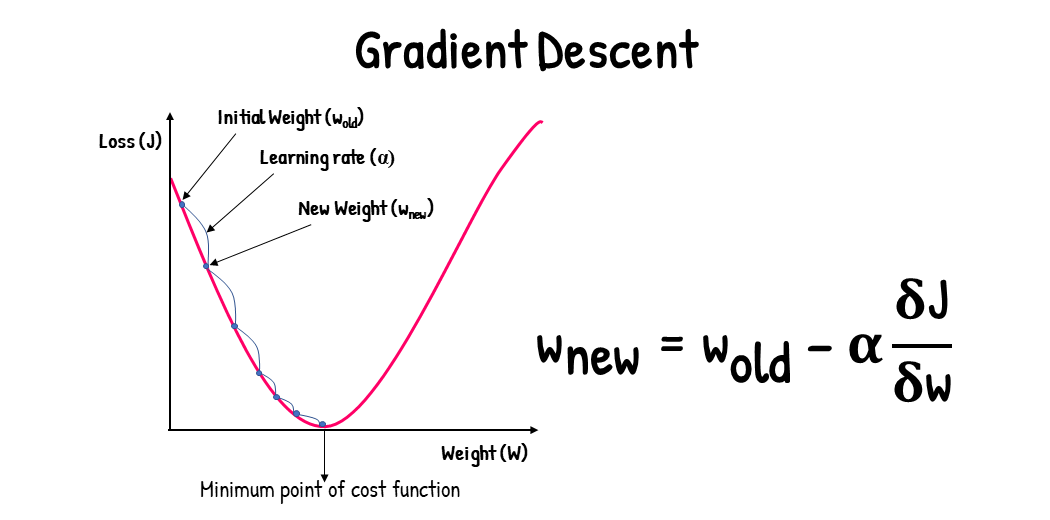
## What is Optimizer

Optimizers are computational algorithms employed to minimize the error function (loss function) or enhance production efficiency. These mathematical functions operate based on the learnable parameters of the model, namely, the weights and biases. The primary role of optimizers is to guide the neural network in adjusting its weights and learning rate systematically to minimize losses. They facilitate the learning process by iteratively updating the model's weights and biases. Well-known optimizers include Stochastic Gradient Descent (SGD), Adam, and RMSprop. Each optimizer is characterized by specific update rules, learning rates, and momentum, contributing to the discovery of optimal model parameters for improved performance.

1. **Type of Optimizer**

### Gradient Desent

Gradient descent is an optimization algorithm designed for convex functions, adjusting its parameters iteratively to minimize a specified function toward its local minimum. This iterative process involves decreasing a loss function by navigating in the direction opposite to the steepest ascent. Relying on the derivatives of the loss function, gradient descent seeks to locate minima. However, it comes with the drawback of utilizing the entire training set data to compute the gradient of the cost function to the parameters. This approach demands a substantial amount of memory and can impede the efficiency of the process.



The Momentum gradient Descent formula update is described as follows:



Advantages of Gradient Descent

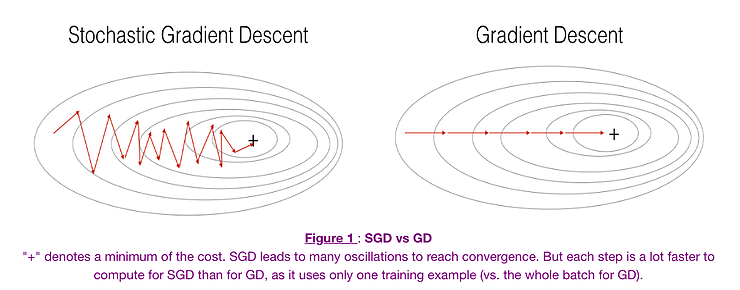
* Easy to understand
* Easy to implement

Disadvantages of Gradient Descent

* Slow convergence, especially for large datasets or complex models.
* Sensitive to the choice of the learning rate.

### Stochastic Gradient Descent (SGD)

Stochastic Gradient Descent (SGD) is a variant of the gradient descent algorithm employed for optimizing machine learning models. It addresses the computational inefficiency associated with gradient descent methods when working with extensive datasets in machine learning endeavors. SGD updates model parameters incrementally. In this method, instead of utilizing the entire dataset for each iteration, a single random instance (or a mini-batch) is chosen to compute the gradient and update the model parameters. This stochastic selection injects randomness into the optimization process, hence the term "stochastic" in stochastic gradient descent. For instance, if the model involves a dataset of 10,000, SGD will perform parameter updates 10,000 times.



Advantages of Stochastic Gradient Descent

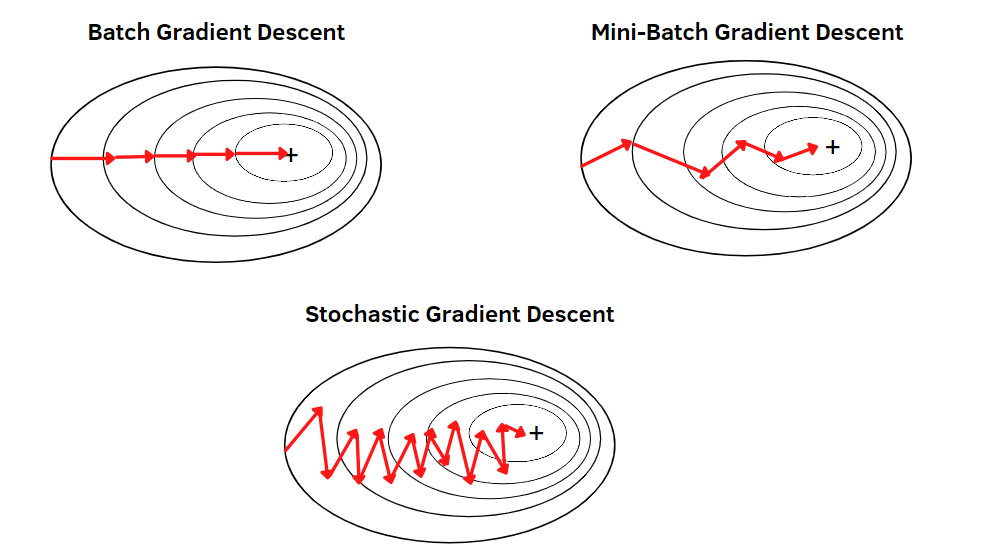
* Frequent updates of model parameter
* Requires less Memory.
* Faster updates as it uses only one training example at a time

Disadvantages of Stochastic Gradient Descent

* The frequent can also result in noisy gradients which may cause the error to increase instead of decreasing it.
* High variance in updates can lead to noisy convergence.
* Frequent updates are computationally expensive.

### Mini-Batch Gradient Descent

Mini-Batch Gradient Descent amalgamates the principles of Stochastic Gradient Descent (SGD) and batch gradient descent by dividing the training dataset into smaller batches and updating the model parameters for each of these batches. This approach strikes a balance between the adaptability of stochastic gradient descent and the efficiency of batch gradient descent. By updating parameters in batches, it helps mitigate variance, leading to more stable convergence. The dataset is typically split into batches containing 50 to 256 examples, selected randomly during each iteration



Advantages of Mini Batch Gradient Descent:

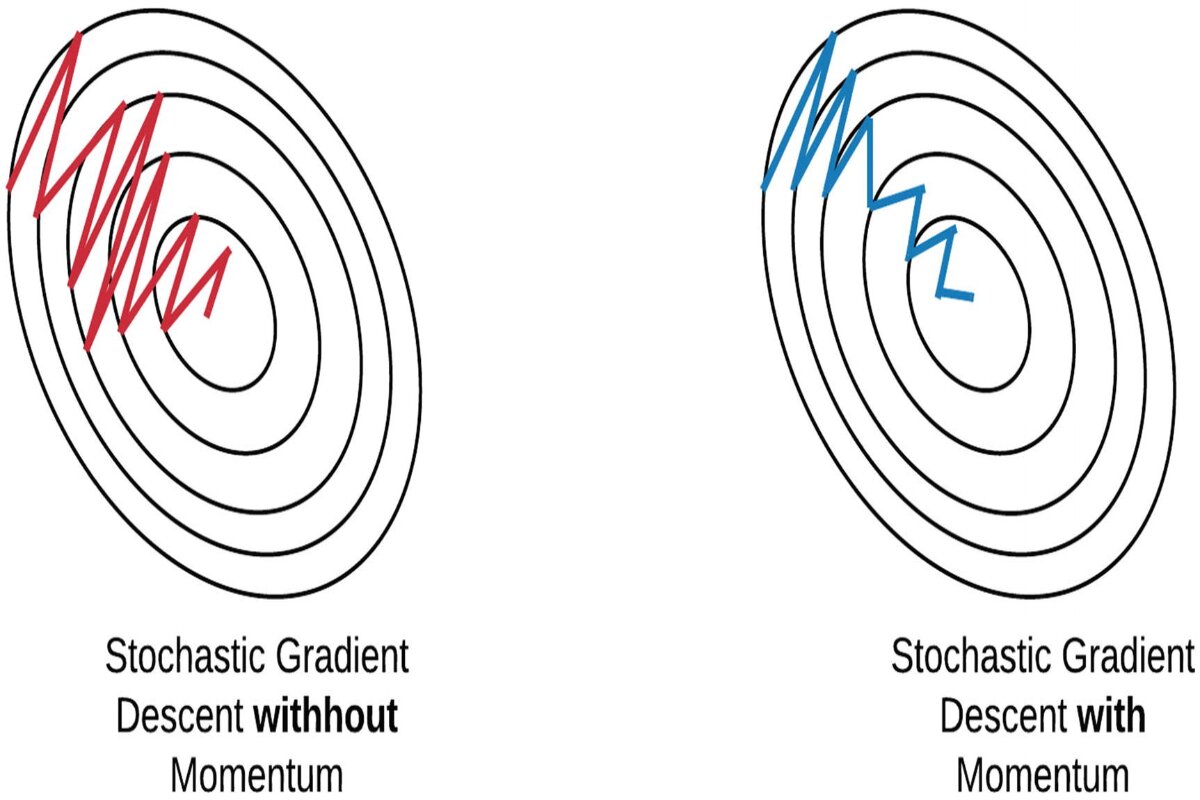
* It leads to more stable convergence.
* more efficient gradient calculations.
* Requires less amount of memory.

Disadvantages of Mini Batch Gradient Descent

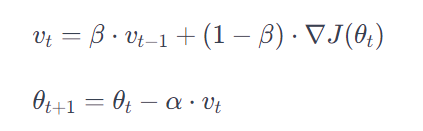
* Mini-batch gradient descent does not guarantee good convergence,
* Requires tuning of the batch size.

### SGD with Momentum

Stochastic Gradient Descent (SGD) with Momentum is a widely adopted optimization algorithm for machine learning model training. It extends the conventional SGD by introducing momentum, a mechanism that enhances convergence speed. This concept mimics the inertia observed in moving objects, preserving the direction of the previous update to some degree. While incorporating the current gradient for fine-tuning, momentum provides increased stability, enabling faster learning and the ability to escape local optima. This approach accelerates the optimization process by leveraging the accumulated information from past updates.



The Momentum gradient Descent formula update is described as follows:



Advantages of SGD with momentum

* Momentum helps to reduce the noise.
* Exponential Weighted Average is used to smoothen the curve.
* Faster convergence due to the momentum term
* Avoid falling into local minimum points

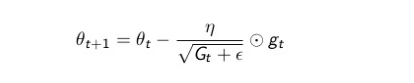
Disadvantage of SGD with momentum

* Extra hyperparameter is added.
* Requires tuning of momentum parameter

### AdaGrad(Adaptive Gradient Descent)

The adaptive gradient descent algorithm distinguishes itself from other gradient descent methods by employing varying learning rates for each iteration. The adjustment in the learning rate is contingent upon the disparity in parameters observed during training. As parameters undergo more significant changes, the learning rate undergoes proportionately smaller adjustments. This adaptation proves advantageous, especially considering that real-world datasets often comprise a mix of sparse and dense features. Adagrad, the algorithm in question, utilizes the following formula for weight updates.

The Adaptive Gradient Descent formula update is described as follows:



Advantages of AdaGrad

* Learning Rate changes adaptively with iterations.
* It is able to train sparse data as well.
* Adaptive learning rates for each parameter

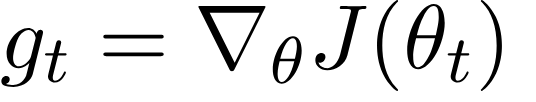
Disadvantage of AdaGrad

* If the neural network is deep the learning rate becomes very small number which will cause dead neuron problem.
* Accumulation of squared gradients may cause the learning rate to diminish too quickly.

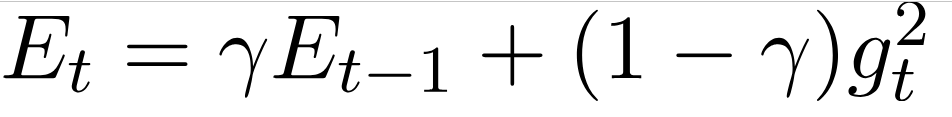
### RMS-Prop (Root Mean Square Propagation)

RMS-Prop (Root Mean Squared Propagation) is an optimization algorithm that dynamically adjusts the learning rate during the training of neural networks. It builds upon the Adaptive Gradient Algorithm, aiming to significantly decrease the computational resources required for training. The algorithm achieves this by exponentially reducing the learning rate whenever the squared gradient falls below a specified threshold. This rapid reduction in the learning rate for small gradients allows RMSProp to smoothly adapt the learning rates for each network parameter. Consequently, RMSProp offers improved performance compared to conventional Gradient Descent methods alone.

Compute the gradient of the objective function with respect to the parameters:



Update the exponentially weighted average of the squared gradients:



Update the parameters



Advantages of RMS-Prop

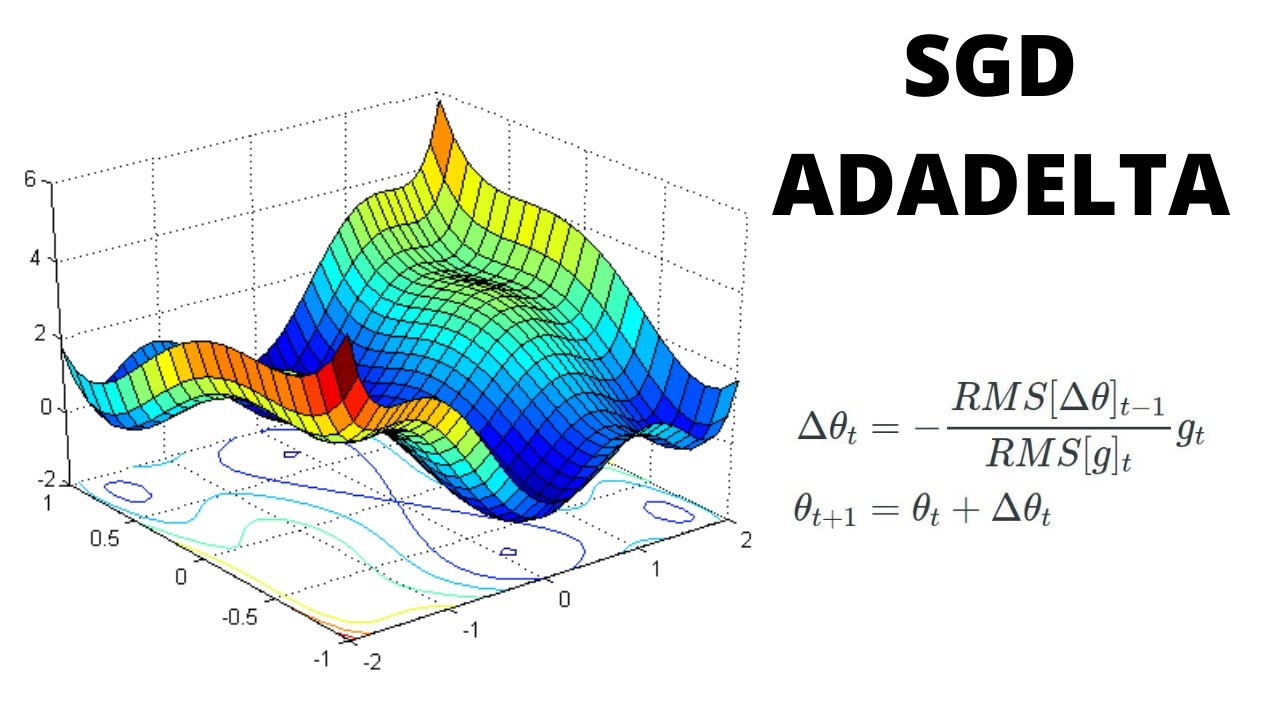
* In RMS-Prop learning rate gets adjusted automatically and it chooses a different learning rate for each parameter.
* Adaptive learning rates based on moving averages.

Disadvantages of RMS-Prop

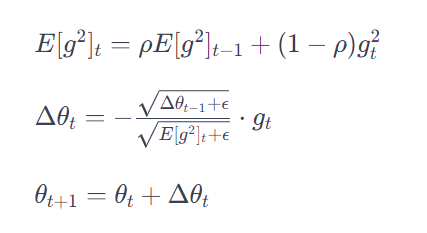
* Slow Learning
* Requires tuning of hyperparameters.

### AdaDelta

Adadelta is an enhancement of Adagrad, specifically addressing its tendency to aggressively and monotonically decrease the learning rate, as well as mitigating the issue of a diminishing learning rate. In contrast to Adagrad, Adadelta dispenses with the use of a fixed learning rate, replacing it with an exponential moving average of the squared deltas, which represent the differences between the current and updated weights. By doing so, Adadelta aims to overcome the challenges associated with a decreasing learning rate, contributing to improved optimization in training models.



The Adaptive Gradient Descent formula update is described as follows:



Advantages of Adadelta

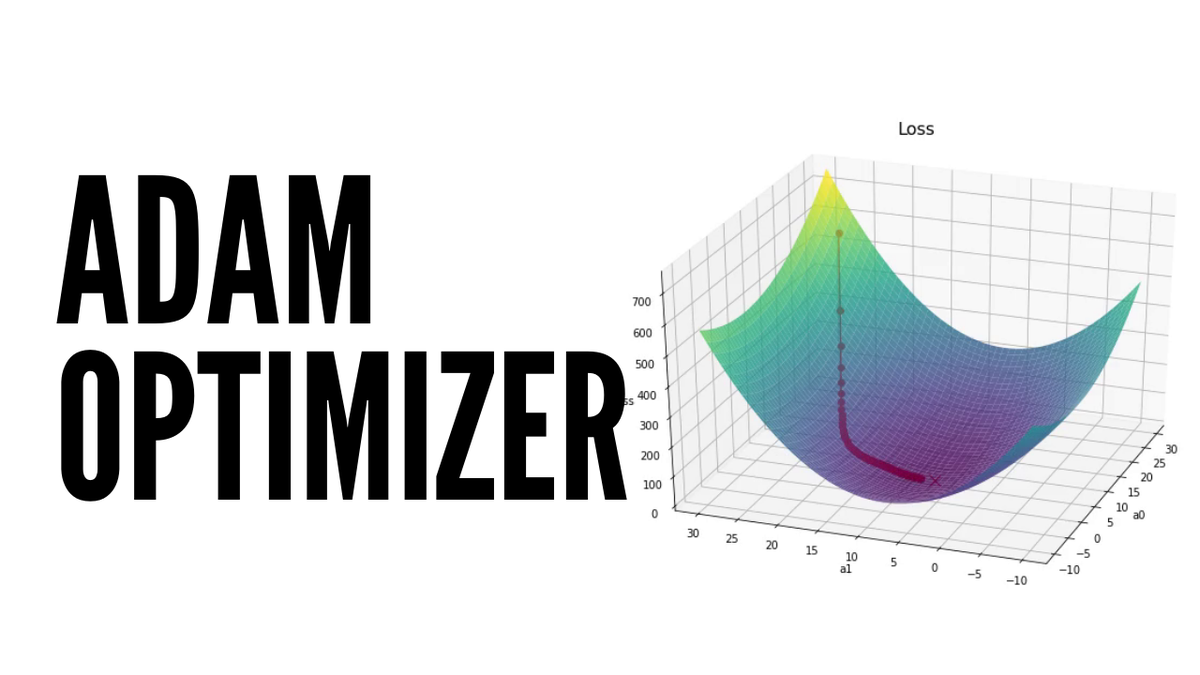
* The main advantage of AdaDelta is that we do not need to set a default learning rate.
* Eliminates the need for a learning rate parameter

Disadvantages of Adadelta

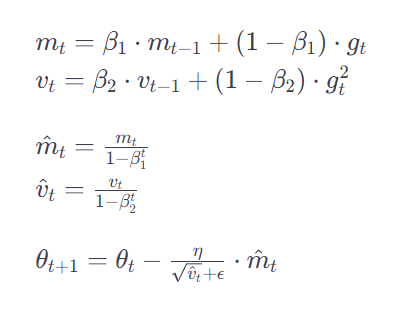
* Computationally expensive

### Adam(Adaptive Moment Estimation)

The Adam optimizer stands out as one of the widely acclaimed gradient descent optimization algorithms. Renowned for its popularity, it is distinguished by its ability to compute adaptive learning rates for individual parameters. Adam maintains two crucial components: the decaying average of previous gradients, akin to momentum, and the decaying average of past squared gradients, reminiscent of RMS-Prop and Adadelta. This unique combination allows Adam to harness the strengths of both methodologies, contributing to its effectiveness in optimizing neural network training.



The Adaptive Moment Estimation formula update is described as follows:



Advantages of Adam:

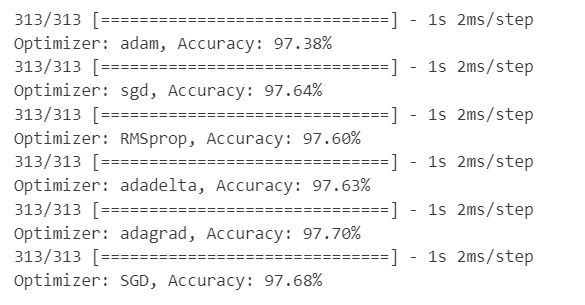
* Easy Implementation
* Requires less memory
* Computationally efficient

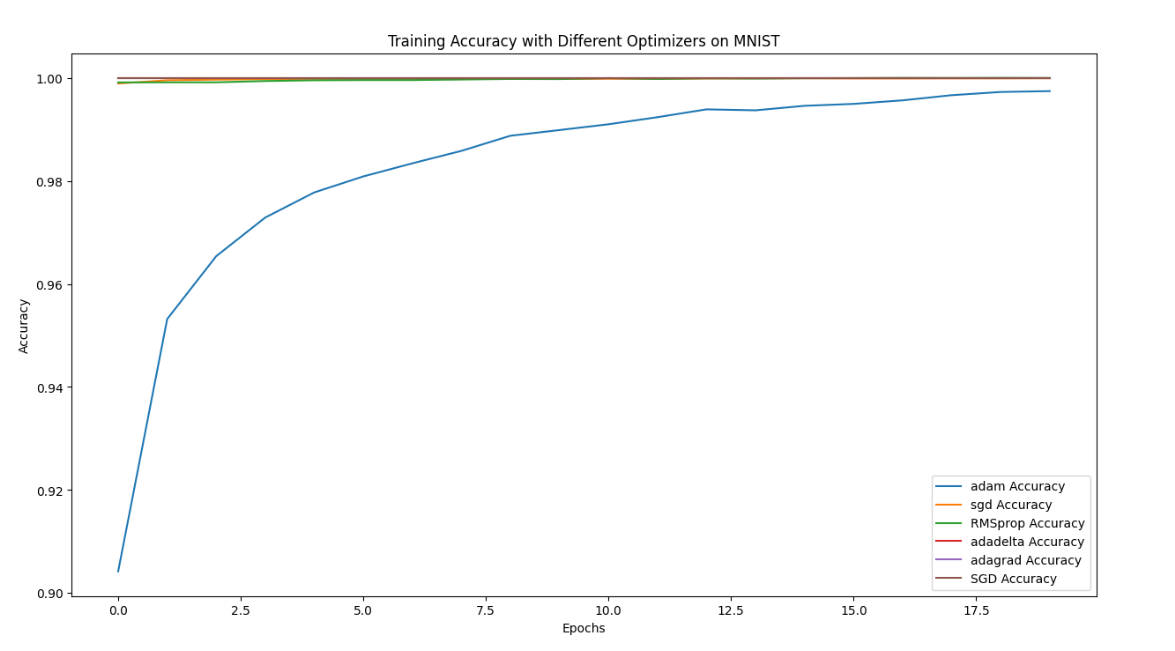
Disadvantages of Adam:

* Can have weight decay problem
* Sometimes may not converge to an optimal solution

1. **Compare Optimizer**

Below is an example of comparing the optimizers learned in the above section about the accuracy of each optimizer and using the MNIST data set and presented in graph form.

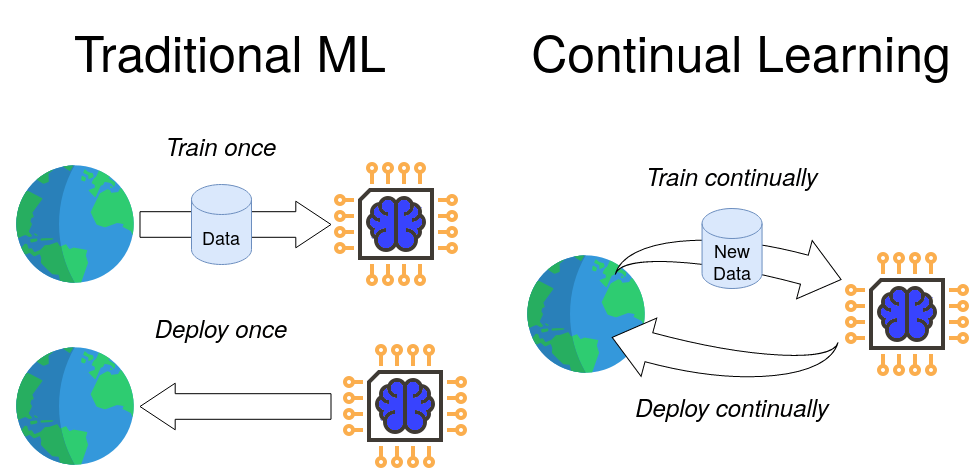




# CHAPTER 2: CONTINUAL Learning and Test Production

## Continual learning

Continual learning, also known as incremental learning or lifelong learning, aims to create adaptive machine learning models capable of continuously acquiring and building upon knowledge. Unlike traditional models with static information, continual learning models can update and improve their skills over time, incorporating new data and tasks without forgetting previously learned knowledge.

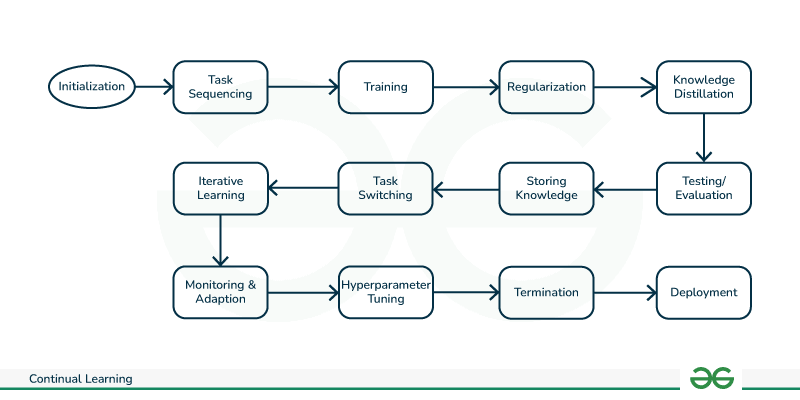


This ongoing learning process mirrors how humans acquire knowledge, constantly expanding their existing understanding. However, a major challenge in continual learning is catastrophic forgetting, where new information overwrites or disrupts prior learning. Mitigating this issue is crucial for maintaining the relevance and efficiency of AI systems in a continuously evolving world.

### Types of Continual Learning

* **Task-based Continual Learning**: This approach involves teaching a version a series of discrete duties over time. The objective of the model is to adapt to every new task while maintaining awareness of previously discovered responsibilities. This class includes methods such as Progressive Neural Networks (PNN) and Elastic Weight Consolidation (EWC).
* **Class-incremental Learning**: The specialty of class-incremental mastering is handling new classes or informational classes over time while maintaining comprehension of previously seen lessons. This is typical of packages such as image recognition, where new object training is added on a regular basis. For class-incremental mastering, techniques such as iCaRL (Incremental Classifier and Representation Learning) are employed.
* **Domain-incremental Learning**: Domain-incremental learning is about adjusting to new domain names or record distributions. In autonomous robotics, for instance, a robot might wish to adjust to various surroundings. To address this situation, area-incremental learning and area variation techniques are applied.

### Process of Continual Learning



1. **Initialization:**
   * Initialize the model with an initial set of parameters. This could involve training the model on an initial dataset or using a pre-trained model as a starting point.
2. **Task Presentation:**
   * Present the model with a task or a batch of data from a particular task. This could involve classifying images, predicting sequences, or any other machine learning task.
3. **Training on the Current Task:**
   * Train the model on the current task using the presented data. This involves updating the model's parameters to minimize a loss function that measures the difference between the model's predictions and the actual outcomes.
4. **Memory Update:**
   * If the model employs a memory mechanism, update the memory with information relevant to the current task. This could involve storing important parameters or examples to avoid catastrophic forgetting.
5. **Task Transition:**
   * Identify when the model is transitioning to a new task or encountering new data. This is crucial for managing task boundaries and adjusting the learning process accordingly.
6. **Adaptation to New Tasks:**
   * When transitioning to a new task, adapt the model to the new data or task requirements. This adaptation might involve updating some of the model's parameters while preserving knowledge from previous tasks.
7. **Regularization to Prevent Catastrophic Forgetting:**
   * Apply regularization techniques to prevent catastrophic forgetting. These techniques penalize changes to important parameters that were crucial for previous tasks, helping the model retain knowledge.
8. **Evaluation on Previous Tasks:**
   * Periodically evaluate the model's performance on previously learned tasks to ensure that the continual learning process does not lead to significant degradation in performance on earlier tasks.
9. **Repeat the Process:**
   * Iterate through steps 2 to 8 for each new task or set of data. This iterative process allows the model to accumulate knowledge and adapt to a variety of tasks over time.
10. **Termination:**
    * The process can be terminated when the model has learned from a sufficient number of tasks or when a specific criterion is met. In some cases, continual learning is an ongoing process to adapt to a continually changing environment**.**

### Key aspects and challenges

1. **Catastrophic Forgetting:** This is a common challenge in continual learning where a model forgets previously learned information when exposed to new data. It occurs because traditional machine learning models are trained to minimize the error on the current task, which may lead to the degradation of performance on previous tasks.
2. **Task Boundaries:** In continual learning, data is often organized into tasks, and the model needs to be able to recognize when a new task is encountered. Adapting to new tasks without forgetting old ones is a significant research challenge.
3. **Memory and Replay Mechanisms:** To mitigate catastrophic forgetting, researchers have explored techniques such as memory-augmented networks, which store important information from previous tasks, and replay mechanisms, where the model is periodically exposed to old data during training.
4. **Regularization Techniques:** Regularization methods can be employed to prevent the model from overfitting to the current task and help retain knowledge from previous tasks. Examples include elastic weight consolidation (EWC) and synaptic intelligence (SI).
5. **Gradient-based Approaches:** Gradient-based approaches involve modifying the learning algorithm to give importance to certain parameters or examples that are important for previous tasks. This helps in preserving the knowledge learned earlier.
6. **Ensemble Methods:** Maintaining an ensemble of models trained on different tasks can help in avoiding catastrophic forgetting. Each model in the ensemble is responsible for a specific task, and they collectively contribute to the overall learning process.
7. **Meta-learning:** Meta-learning involves training models in a way that they can quickly adapt to new tasks with minimal data. This concept is often used in continual learning to improve the model's ability to handle new information.
8. **Online and Batch Learning:** Continual learning can be approached using online learning, where the model is updated with new data as it arrives, or batch learning, where data is collected and used in batches. The choice depends on the specific application requirements.

## Test Production

Test production in machine learning typically refers to the deployment and evaluation of machine learning models in a production environment. It involves taking a trained model that has been tested on historical data and putting it into use to make predictions or classifications on new, real-world data.

### Key components of test production

**Performance Testing**:

* Performance testing is a crucial evaluation process for a machine learning model, focusing on its speed, resource utilization, and scalability. This aims to verify that the model meets operational requirements by assessing its prediction speed, computational resource demands, and ability to scale with increasing data volume or user load.

**A/B Testing**:

* A/B testing, also known as split testing, involves comparing two versions (A and B) of a model to determine their relative performance based on predefined metrics. Through randomized user or data assignment, A/B testing provides insights into which model version yields superior outcomes, aiding decision-making in model deployment or updates.

**Interpretability**:

* Interpretability in machine learning pertains to the model's transparency, enabling users to comprehend and explain the rationale behind its predictions. This attribute fosters trust, assists in compliance with regulations, and aids in debugging by providing insights into the decision-making process.

**Reproducibility:**

* Reproducibility involves the ability to recreate identical results from a previous experiment or study. In the context of machine learning, this encompasses version control for the codebase, replicating the computational environment, and documenting datasets used during model training and testing.

**API Testing:**

* API testing assesses the functionality, reliability, and security of an Application Programming Interface (API). In the realm of machine learning, this includes validating model deployment endpoints, ensuring consistent input-output behavior, and monitoring the overall health and security of the deployed model API.

### Step of the test production process in machine learning

1. **Model Deployment:**
   * Deployment involves integrating the trained model into the production environment. This can be done through various mechanisms, such as deploying as a web service, embedding within an application, or integrating into a larger system.
2. **Data Input and Preprocessing:**
   * Real-world data may have variations, missing values, or other issues not present in the training data. Preprocessing steps, such as handling missing data, scaling features, and encoding categorical variables, must be adapted to the characteristics of the production data.
3. **Scalability and Efficiency:**
   * Production systems may experience fluctuations in data volume and traffic. It's essential to ensure that the deployed model can handle these variations efficiently. This may involve optimizing code, utilizing hardware accelerators, or implementing distributed computing strategies.
4. **Monitoring and Logging:**
   * Implementing a robust monitoring system helps track the model's performance over time. Logging can capture information about input data, predictions, and any anomalies. This data is crucial for diagnosing issues, identifying trends, and making improvements.
5. **Feedback Loop:**
   * Establishing a feedback loop involves collecting data on model predictions and comparing them to actual outcomes. This data is used to retrain the model periodically, ensuring it remains accurate as the underlying data distribution evolves.
6. **Security and Compliance:**
   * Security measures must be in place to protect the model and the data it processes. Compliance with regulations and industry standards (such as GDPR, HIPAA, or others) is critical. This may involve encryption, access controls, and other security practices.
7. **Performance Evaluation:**
   * Regularly evaluating the model's performance helps identify potential issues early on. This evaluation may involve comparing model predictions to ground truth, analyzing metrics like accuracy, precision, recall, and monitoring for any degradation in performance.
8. **Versioning and Rollback:**
   * Version control ensures that different iterations of the model can be tracked. This is crucial for rolling back to a previous version if issues arise with a new deployment. It also facilitates collaboration among teams working on model development and deployment.
9. **Documentation:**
   * Comprehensive documentation is essential for the entire deployment process. This includes documenting the model architecture, training process, preprocessing steps, deployment configurations, and any other relevant information. Documentation aids in troubleshooting, onboarding new team members, and maintaining a clear understanding of the deployed system.
10. **A/B Testing (Optional):**
    * A/B testing involves deploying multiple versions of a model simultaneously and comparing their performance. This can help in making informed decisions about model updates and improvements before fully replacing the existing model.
11. **User Acceptance Testing (UAT):**
    * Before deploying to a broader audience, user acceptance testing involves having a subset of users or stakeholders interact with the model to ensure it meets their expectations and requirements.

# REFERENCE

<https://towardsdatascience.com/progressive-neural-networks-explained-implemented-6f07366d714d#0d70>

<https://www.valamis.com/hub/continuous-learning#:~:text=Continuous%20learning%20is%20the%20process,initiative%20and%20taking%20on%20challenges>.

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

<https://www.geeksforgeeks.org/continual-learning-in-machine-learning/>

<https://medium.com/@istabrak.abbes_65953/what-is-continual-learning-491ed9e3874d>

<https://viblo.asia/p/optimizer-hieu-sau-ve-cac-thuat-toan-toi-uu-gdsgdadam-Qbq5QQ9E5D8>