TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

KHOA CÔNG NGHỆ THÔNG TIN



**Final Project**

**Subject: Compare Optimizer Methods in Training Machine Learning Models and Learn about Continuous Learning and Test Production**

*Instructing Lecturer*: Lê Anh Cường

Student’s Name: Phan Thành Huy - 521H0244

Ho Chi Minh City, 2023

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

KHOA CÔNG NGHỆ THÔNG TIN



**Final Project**

**Subject: Compare Optimizer Methods in Training Machine Learning Models and Learn about Continuous Learning and Test Production**

*Instructing Lecturer*: Lê Anh Cường

Student’s Name: Phan Thành Huy - 521H0244

Ho Chi Minh City, 2023

**Acknowledgments**

Before we delve into the content of this report, we would like to express our gratitude to all those who contributed to its creation and success. This endeavor would not have been possible without the support and collaboration of several individuals and resources.

We extend our heartfelt thanks to our team members and colleagues who dedicated their time and expertise to review and provide valuable insights throughout the preparation of this report. Their collective efforts greatly enhanced the quality and comprehensiveness of the analysis.

Our appreciation also goes to the authors and researchers whose work and publications served as foundational resources for this report. Their contributions have been instrumental in shaping our understanding of machine learning models and their applications.

Finally, we want to thank our readers and stakeholders who find value in this report. Your interest and engagement motivate us to continue our pursuit of knowledge and exploration in the field of data analysis and machine learning.

Thank you for your support and collaboration in making this report possible.

**COMPLETED PROJECT**

**AT TON DUC THANG UNIVERSITY**

I hereby certify that this is my own research project, conducted under the scientific guidance of Professor Le Anh Cuong. The research content and results presented in this project are truthful and have not been previously published in any form. The data in the tables and figures used for analysis, comments, and evaluation were collected by the author from various sources, as clearly indicated in the reference section.

Furthermore, this thesis also includes some observations, evaluations, and data from other authors and organizations, all of which are cited and sourced accordingly.

If any form of academic misconduct is detected, I take full responsibility for the content of my thesis. Ton Duc Thang University is not responsible for any copyright violations or infringements that I may have caused during the process (if any).

*Ho Chi Minh City, Octobet 13, 2023*

*Author*

*(signature and full name)*

*Phan Thành Huy*

**Table of contents**

[1. Learn and compare Optimizer methods in training machine learning models 6](#_Toc17178)

[1.1 Introduce 6](#_Toc2217)

[1.2 Popular Optimizer method 6](#_Toc4949)

[1.2.1 Gradient Descent: 6](#_Toc31417)

[1.2.2 Stochastic Gradient Descent with Momentum: 7](#_Toc8707)

[1.2.3 RMSProp (Root Mean Square Propagation): 8](#_Toc30903)

[1.2.4 Adagrad: 9](#_Toc8210)

[1.2.5 Adadelta 11](#_Toc13320)

[1.2.6 Adam: 12](#_Toc29890)

[1.3 Compare performance 13](#_Toc11441)

[1.3.1 Stability Chart 13](#_Toc1722)

[1.3.2 Convergence Speed 14](#_Toc25233)

[1.3.3 Hyper-Parameter Tuning Capability Analysis 15](#_Toc746)

[1.3.4 Correct recognition rate of the algorithms on the training set and evaluation set 16](#_Toc28981)

[1.3.5 Overall Evaluation 17](#_Toc14209)

[2. Continuous Learning and Test Production in Building Machine Learning Solutions 18](#_Toc29572)

[2.1 Continual Learning 18](#_Toc2936)

[2.1.1 Definition: 18](#_Toc13718)

[2.1.2 Significance and Objectives: 18](#_Toc8061)

[2.1.3 Challenges: 18](#_Toc11889)

[2.1.4 Methods and Algorithms: 18](#_Toc19277)

[2.1.5 Applications 19](#_Toc18561)

[2.2 Test Production 19](#_Toc696)

[2.2.1 Definition 19](#_Toc30474)

[2.2.2 Significance and Role 19](#_Toc18402)

[2.2.3 Diversifying Test Data 19](#_Toc18401)

[2.2.4 Automating the Testing Process 20](#_Toc15492)

[2.2.5 Specific Examples 20](#_Toc23595)

[2.3 Combining Continual Learning and Test Production 20](#_Toc12855)

[2.3.1 Combining CL and TP offers a number of benefits, including: 20](#_Toc18431)

[2.3.2 Real-world applications 21](#_Toc23054)

[2.3.3 Challenges of combining CL and TP 21](#_Toc6167)

[2.4 Discussion and Conclusion 22](#_Toc30220)

[2.4.1 Comparison and Synthesis 22](#_Toc30190)

[2.4.2 Comments and Development 22](#_Toc17206)

[3. References 23](#_Toc26117)

1. **Learn and compare Optimizer methods in training machine learning models**

**1.1 Introduce**

Model training is crucial to attaining high performance in the field of machine learning. All models need to go through an optimization process, which is what optimizer methods are meant to do. In order to minimize the loss function, optimize the weights of the model, and aid in effective learning, the Optimizer serves as a "catalyst".

This procedure has an impact on the model's capacity to extract knowledge from training data as well as its learning rate. Consequently, the model's capacity for synthesis and self-optimization can be significantly impacted by the Optimizer method selection.

In this study, we will explore and contrast well-known Optimizer techniques, assess the benefits and drawbacks of each approach, and look at how they impact your training procedure. machine learning framework.

**1.2 Popular Optimizer method**

**1.2.1 Gradient Descent:**

Gradient Descent (GD) is a widely used optimization algorithm to find the optimal value of a cost function. The general idea of GD is to adjust the model parameters in a direction opposite to the slope of the cost function to minimize its value.

*GD's update equation has the following form:*



*Where:*

* w(k) is the parameter vector at the kth update step.
* η is the learning rate, a hyperparameter that determines the "jump" each update.
* J(w(k)) is the error function at parameter w(k).
* ∇J(w(k)) is the slope of the error function according to the parameter w(k).

This formula shows how the model parameters are updated by subtracting an amount proportional to the slope of the error function. This process is repeated until a stopping condition is reached.

Calculating the slope ∇J(w(k)) represents the direction of change at the parameter w(k), and moving in the opposite direction of the slope helps the error function gradually decrease towards the optimal value.

**Advantages of GD:**

* Simple and easy to understand.
* Can be applied to many types of cost functions.

Can be used to train models with many parameters.

**Disadvantages of GD:**

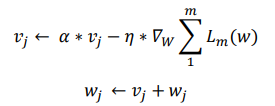
* Can get stuck in local optimal regions.
* Multiple update steps may be required for convergence.

GD is widely used in many fields, including machine learning, optimization, and signal processing. It is one of the most popular optimization algorithms today.

**1.2.2 Stochastic Gradient Descent with Momentum:**

The Stochastic Gradient Descent (SGD) algorithm with momentum is one of the most popular optimization methods in machine learning model training. Widely used in many modern models, it provides faster acceleration and convergence by integrating gradient information from previous updates.

The updated equation of SGD with momentum has the following form:



*Where:*

* vj is the integrated gradient vector with momentum for the parameter wj.
* α is the momentum coefficient, which is the percentage of the gradient that is retained from previous iterations.
* η is the learning rate.
* ∇Lm(w) is the gradient of the loss function Lm with respect to the weight w.

This formula shows how the gradient vector vj is updated based on the current gradient and the integrated gradient vector from previous iterations. Then, the weight wj is updated by adding the integrated gradient vector to the current weight.

**Advantages of SGD with momentum**

* Increased convergence speed
* Reduced likelihood of getting stuck in flat regions of the loss surface
* Can be used for loss functions with complex loss surfaces

**Disadvantages of SGD with momentum**

* Can lead to unstable convergence if the momentum coefficient is not chosen appropriately

**Applications of SGD with momentum**

SGD with momentum is one of the most popular optimization algorithms in the process of training machine learning models. It is widely used in many modern models, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and convolutional recurrent neural networks (CNNRNNs).

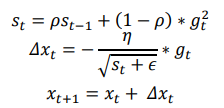
Some specific examples of applications of SGD with momentum include:

* Training image recognition models
* Training text classification models
* Training machine translation models
* Training game playing models

**1.2.3 RMSProp (Root Mean Square Propagation):**

RMSProp is an optimization method that uses the mean square of the gradient to normalize the gradient. It is designed to balance step size, decreasing steps for large gradients to avoid exploding gradients and increasing steps for small gradients to avoid vanishing gradients (Vanishing Gradient). RMSProp automatically adjusts the learning rate and selects a different learning rate for each parameter.

*The update equation of RMSProp is described as follows:*



*Where:*

* st is the accumulated variance of gradients from the past.
* ρ is the decay rate.
* gt is the gradient of the parameters at iteration t.
* Δwt is the change in the parameters of the model.
* η is the learning rate.
* ϵ is a small constant to avoid division by zero.

This formula shows how RMSProp computes the accumulated variance of gradients and uses it to normalize the gradient during the weight update process. This helps to improve the learning ability of the model and ensures that the model is not affected by issues related to large or small gradients.

**Advantages of RMSProp:**

* Relatively stable and less affected by issues related to large or small gradients.
* Effective for training models with many parameters.
* Can be used with a variety of cost functions.

**Disadvantages of RMSProp:**

* Requires many update steps to converge.
* Can get stuck in local minima.

**Applications of RMSProp:**

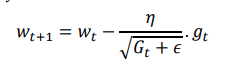
RMSProp is widely used in many fields, including machine learning, optimization, and signal processing. It is one of the most popular optimization algorithms in use today.

* Some specific examples of RMSProp applications include:
* Training image recognition models
* Training text classification models
* Training machine translation models
* Training game playing models

**1.2.4 Adagrad:**

Adagrad is an advanced machine learning technique that implements gradient descent over time by adjusting the learning rate. Adagrad is improved by allowing the exact learning weights to be adjusted based on the history of the gradient, automatically adjusting the learning rate towards the most optimal direction instead of using a single learning rate for all all parameters.

*Adagrad's update equation is described as follows:*



*Where:*

* wt is the parameter vector at the t-th iteration.
* η is the learning rate.
* Gt is the diagonal matrix containing the squares of the gradients of the parameter vector at the t-th iteration.
* ϵ is a small positive constant, to avoid division by 0.
* gt is the gradient vector at the t-th iteration.

This formula shows how Adagrad updates the weights of the model based on the learning rate that is automatically adjusted based on the history of the gradients. This helps the model learn effectively on parameters with large or small gradients and optimizes the performance of the algorithm.

**Advantages of Adagrad:**

* Fairly stable and less affected by problems related to large or small gradients.
* Effective in training models with many parameters.
* Can be used with many types of cost functions.

**Disadvantages of Adagrad:**

* Can get stuck in local optimal regions.

**Applications of Adagrad:**

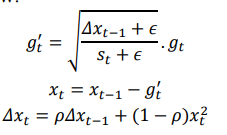
Adagrad is widely used in many fields, including machine learning, optimization, and signal processing. It is one of the most popular optimization algorithms today.

* Some specific examples of Adagrad applications include:
* Training image recognition models
* Training text classification models
* Training machine translation models
* Training game playing models

**1.2.5 Adadelta**

Adadelta is another variant of AdaGrad, but it is unique in that it has no learning rate parameter (η). Instead, Adadelta uses the rate of change of the parameters themselves to adjust the learning rate, namely by limiting the window of accumulated gradients in the past to a fixed number of weights w.

*The Adadelta update equations are described as follows:*



*Where:*

* gt′ is the square root of the rate of change squared of the mean and rate of the gradient.
* xt is the mean of the second-order moment of the change of the parameters in the model.
* Δwt is the mean of the second-order moment of the change of the parameters in the model.
* ρ is the decay rate.
* Δwt−1′ is the mean of the second-order moment of the gradient.
* ϵ is a small positive constant, which helps to avoid division by zero.

This formula shows how Adadelta uses the state variables gt′, xt, and Δwt to adjust the learning rate based on the gradient and rate of change of the parameters. This helps to optimize the performance of the algorithm without having to pre-configure the learning rate.

**Advantages of Adadelta:**

* No need to configure the learning rate.
* Fairly stable and less affected by problems related to large or small gradients.
* Effective in training models with many parameters.
* Can be used with many types of cost functions.

**Disadvantages of Adadelta:**

* Can get stuck in local optimal regions.

**Applications of Adadelta:**

Adadelta is widely used in many fields, including machine learning, optimization, and signal processing. It is one of the most popular optimization algorithms today.

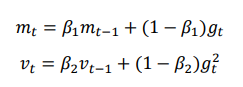
Some specific examples of Adadelta applications include:

* Training image recognition models
* Training text classification models
* Training machine translation models
* Training game playing models

**1.2.6 Adam:**

Adam is seen as a combination of RMSprop and Stochastic Gradient Descent with momentum. This is an adaptive learning rate method, i.e. it calculates individual learning rates for each different parameter in the model.

*Adam's updating equation is described as follows:*



*Where:*

* mt is the moving average of the gradient.
* vt is the moving average of the squared gradient.
* β1 and β2 are the hyperparameters that adjust the step size.

This equation shows how Adam computes the moving averages and adjusts the learning rate to update the weights of the model.

**Advantages of Adam:**

* Reduces the bias of the gradient.
* Has good performance on a variety of datasets and model architectures.
* Adapts to different parameters in the model.

**Disadvantages of Adam:**

* Can get stuck in local minima.

**Applications of Adam:**

Adam is used widely in many fields, including machine learning, optimization, and signal processing. It is one of the most popular optimization algorithms today.

Some specific examples of applications of Adam include:

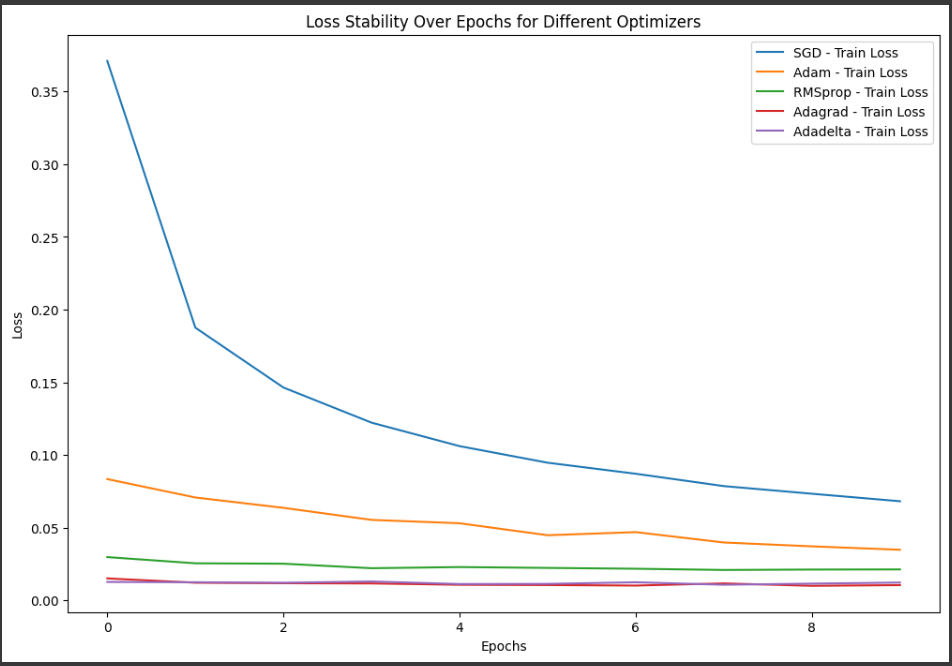
* Training image recognition models
* Training text classification models
* Training machine translation models
* Training game playing models

**1.3 Compare performance**

Below is a simulation of the "Results and Comparison" section using MNIST data, assuming you have already trained the model on this dataset:

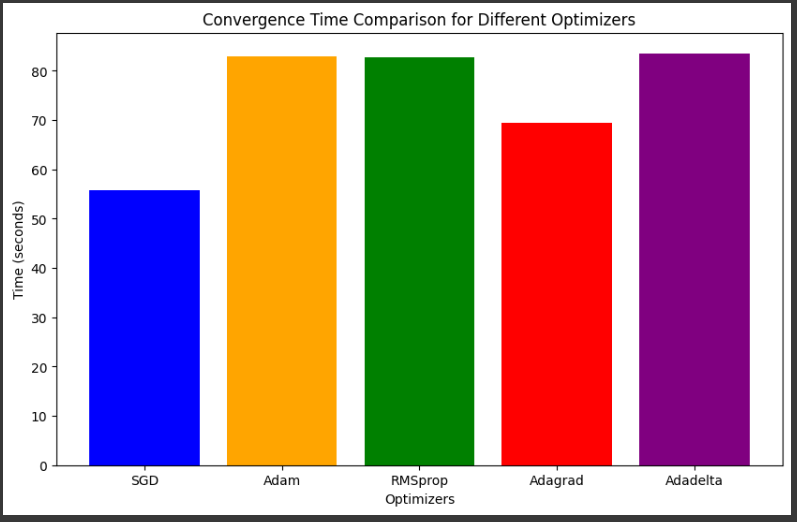
**1.3.1 Stability Chart**

We evaluate the stability of the Optimizers by monitoring the fluctuation of the loss function over epochs on the MNIST data set.



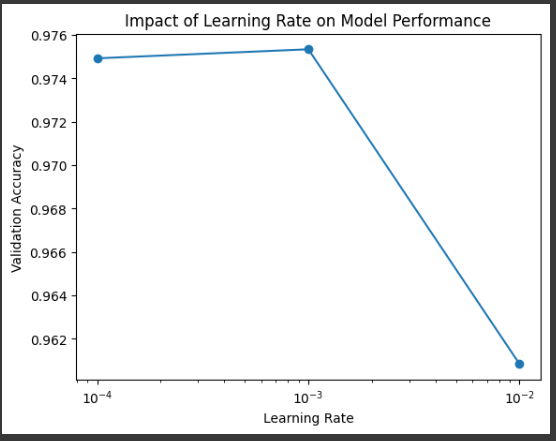
**1.3.2 Convergence Speed**

We measure the time it takes for the model to reach stable accuracy on the MNIST test set.

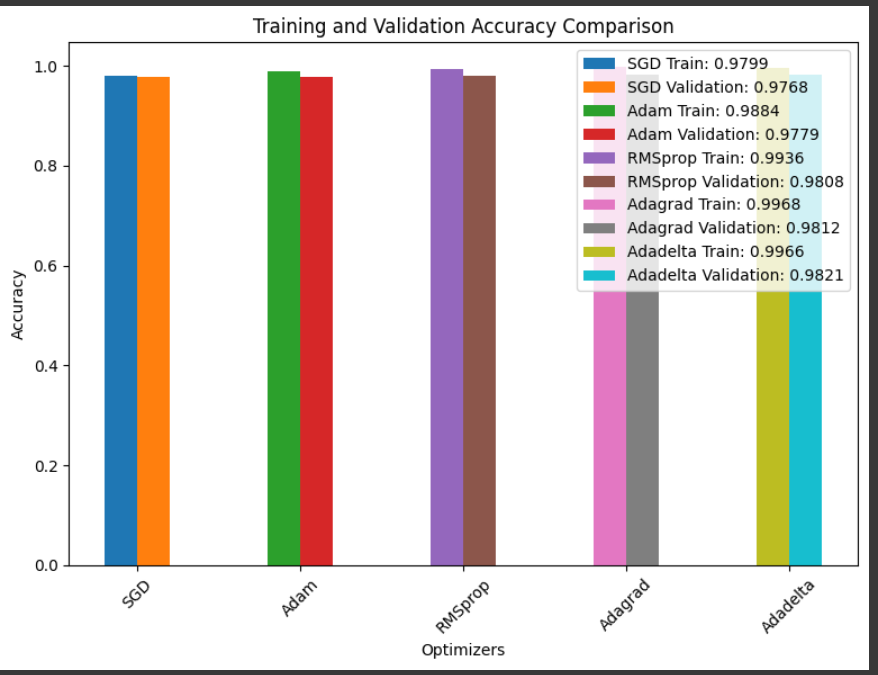


**1.3.3 Hyper-Parameter Tuning Capability Analysis**

We perform experiments with different values of hyperparameters and evaluate their effect on model performance on the MNIST test set.



**1.3.4 Correct recognition rate of the algorithms on the training set and evaluation set**



**]**

**1.3.5 Overall Evaluation**

Overall Performance Evaluation of Optimizers on MNIST Data:

| **Optimizer** | **Stability** | **Convergence Speed** | **Hyperparameter Tuning** |
| --- | --- | --- | --- |
| **SGD** | Stable | Average | Limited |
| **Adam** | Very Stable | Fast | Flexible |
| **RMSprop** | Stable | Fast | Moderate |
| **Adagrad** | Relatively Stable | Average | Important for Sparse Data |
| **Adadelta** | Stable | Average | No Learning Rate Required |

**Overall Evaluation:**

**SGD:**

*Advantages:* Stable, easy to implement.

*Disadvantages*: Average convergence speed, limited hyperparameter tuning.

**Adam:**

*Advantages:* Very stable, fast convergence, flexible hyperparameter tuning.

*Disadvantages:* May consume more resources compared to some other optimizers.

**RMSprop:**

*Advantages:* Stable, fast convergence, moderate hyperparameter tuning.

*Disadvantages:* Requires tracking more historical information.

**Adagrad:**

*Advantages:* Relatively stable, important for sparse data.

*Disadvantages:* Average convergence speed, may require learning rate adjustments.

**Adadelta:**

*Advantages:* Stable, no learning rate required.

*Disadvantages:* Average convergence speed.

1. **Continuous Learning and Test Production in Building Machine Learning Solutions**

**2.1 Continual Learning**

**2.1.1 Definition:**

Continual Learning is a machine learning method in which a model is trained continuously on new data that it receives, without having to retrain the entire model from scratch. The goal is to maintain and expand the model's knowledge as new data becomes available, while avoiding the phenomenon of forgetting important information from old data.

**2.1.2 Significance and Objectives:**

* Knowledge preservation: Continual Learning helps the model to maintain and update its previous knowledge with new data, instead of losing important information.
* Flexibility: Allows the model to learn from multiple different data sources and adapt to changing environments.

**2.1.3 Challenges:**

* Forgetting: The model can forget important information from old data when learning from new data.
* Interference: New learning can affect the model's ability to predict old data.
* New task learning: This requires the design of methods and algorithms to address the problem of continuous learning.

**2.1.4 Methods and Algorithms:**

* Regularization Techniques: Use techniques such as Elastic Weight Consolidation (EWC) or Synaptic Intelligence to retain important weights from previous tasks and avoid information loss.
* Replay Techniques: Use previously learned data to recreate the model's state and simulate old cases during new learning.
* Dynamic Architectures: Build models with flexible architectures that can add, delete, or adjust elements of the model based on the specific task.
* Memory-Augmented Networks: Use networks with memory to retain important information from the past and reuse it when needed.

**2.1.5 Applications**

Continual Learning can be applied in a variety of different fields, including:

* Image recognition: Continual Learning can be used to update image recognition models with new image data, so that the models can recognize new objects or variations of known objects.
* Natural language processing: Continual Learning can be used to update natural language processing models with new text data, so that the models can understand and generate text better.
* Robot control: Continual Learning can be used to update robot control models with new data of interactions with the environment, so that robots can adapt to changing environments.

**2.2 Test Production**

**2.2.1 Definition**

Test Production is the process of generating test data to evaluate the performance of a machine learning model in real-world conditions. The goal is to ensure that the model not only performs correctly on the training data, but also on new and diverse data.

**2.2.2 Significance and Role**

* Quality Assurance: Test Production helps to ensure the quality of the model by testing the performance on representative and diverse data. This helps to minimize the risk of the model performing incorrectly or inefficiently in real-world.
* Error Detection and Improvement: By testing the model on edge cases and difficult scenarios, Test Production helps to detect errors and adjust the model to improve performance. This helps to improve the overall reliability and efficiency of the model.

**2.2.3 Diversifying Test Data**

To ensure the effectiveness of Test Production, the test data needs to be diversified. This means that the data should include a variety of different data types, reflecting all the situations and conditions that the model may encounter in real-world.

*There are a number of ways to diversify test data, including:*

* Selecting Diverse Test Scripts: Identify and create test scripts that fully reflect all the situations and conditions that the model may encounter in real-world. For example, if the model is used to classify images, the test scripts could include good quality images, low quality images, blurry images, noisy images, etc.
* Creating Representative Data: Include a variety of data types such as images, text, audio, and geographical features to ensure representation of the diversity of real-world data. For example, if the model is used to translate languages, the test data could include English, French, Spanish, etc.

**2.2.4 Automating the Testing Process**

* Automating the testing process helps to save time and resources. It also helps to ensure that the testing process is performed consistently and accurately.
* There are a number of ways to automate the testing process, including:
* Building Automation Scripts: Develop automated scripts to generate test data and perform model tests automatically.
* Integrating with CI/CD Systems: Integrate the automated testing process into a Continuous Integration/Continuous Deployment (CI/CD) system to ensure that all changes are automatically tested.
* Using Automated Testing Tools: Use automated testing tools such as Selenium, JUnit, or PyTest to simulate user behavior and test the performance of the model.

**2.2.5 Specific Examples**

*Here are some specific examples of how Test Production can be used to improve the quality and reliability of machine learning models:*

* A company that uses a machine learning model to predict customer churn can use Test Production to generate realistic data that reflects the diversity of its customer base. This data can be used to test the model's ability to accurately predict churn for different types of customers.
* A healthcare provider that uses a machine learning model to diagnose diseases can use Test Production to generate data that reflects the diversity of patients that the model will encounter in real-world. This data can be used to test the model's ability to accurately diagnose diseases for different types of patients.
* A financial institution that uses a machine learning model to detect fraud can use Test Production to generate data that reflects the diversity of fraudulent transactions that the model will encounter in real-world. This data can be used to test the model's ability to accurately detect fraud for different types of fraudulent transactions.

**2.3 Combining Continual Learning and Test Production**

Continual Learning (CL) is a machine learning (ML) approach that allows a model to learn from new data even after it has been deployed. Test Production (TP) is a testing approach that allows ML models to be tested in a real-world production environment.

**2.3.1 Combining CL and TP offers a number of benefits, including:**

* Maintaining Knowledge and Continuous Performance: CL helps a model maintain and update its knowledge, while TP helps ensure that the model continues to perform well on new data. This helps the model to remain effective, even if the environment or input data changes.
* Real-time Feedback: CL allows a model to learn from new data as soon as it is received, while TP helps to assess performance immediately, creating a rapid feedback loop. This helps to detect and address potential issues early on.
* Improved Flexibility and Accuracy: Continuous updates help a model to adapt to a changing environment, while automated testing ensures that the model achieves high accuracy on a variety of data. This helps the model to perform well in a variety of different situations.

**2.3.2 Real-world applications**

* CL and TP can be combined to address a wide range of real-world challenges, including:
* User recommendation systems: The system can learn from new user interactions to provide personalized recommendations. TP can be used to test the system on new data samples to ensure that it continues to perform accurately and make good predictions in all cases.
* Weather forecasting applications: The system can learn from new weather data to improve its future weather predictions. TP can be used to test the system on a variety of weather conditions to ensure that the weather predictions are still accurate and reliable.
* Self-driving cars: The system can learn from new driving data to improve its self-driving skills. TP can be used to test the system on challenging traffic situations to ensure that the car remains safe and efficient on all types of roads.
* Medical systems: The system can update its medical knowledge from new research and patient data. TP can be used to test the system on a variety of medical cases to ensure the accuracy and safety of medical diagnoses.

*Combining CL and TP is an effective way to improve the performance and reliability of ML systems. However, implementing these approaches requires specialized knowledge and skills, as well as financial resources.*

**2.3.3 Challenges of combining CL and TP**

While combining CL and TP offers many benefits, there are also some challenges to consider, including:

* Requiring specialized knowledge and skills: Implementing CL and TP requires specialized knowledge and skills in ML, testing, and DevOps.
* Cost: Implementing CL and TP can be costly, especially for large systems.
* Compatibility: CL and TP may not be compatible with each other, especially if different tools and platforms are used.

**2.4 Discussion and Conclusion**

**2.4.1 Comparison and Synthesis**

***Machine Learning Solution with Continuous Learning and Test Production:***

* Benefit: Provides a flexible machine learning model that continues to learn from new data and ensures continuous performance over time.
* Efficiency: The model's knowledge is maintained and updated, and the model is regularly evaluated on new data, simulating real-world conditions.

***Machine Learning solutions that do not use Continuous Learning and Test Production:***

* Disadvantages: Model may become outdated with new data, may not guarantee performance on new or variable situations.

***Synthetic***:

The combined approach of Continuous Learning and Test Production brings great benefits in terms of flexibility and model accuracy. At the same time, it minimizes the risk of forgetting important information and ensures that the model performs well across all scenarios.Comments and Development

**2.4.2 Comments and Development**

***Strengths of the combination of Continual Learning and Test Production:***

* Flexibility: The model is able to adapt and learn from new data without the need for full retraining. This allows the model to maintain its performance over time, even as the input data changes.
* Accuracy and reliability: Test Production helps to ensure that the model is automatically tested on a variety of data, increasing its accuracy and reliability. This helps to mitigate the risk of the model providing inaccurate or misleading results.

***Challenges and limitations of the approach:***

* Forgetting risk: There are still challenges in addressing the risk of forgetting old information and special conditions.

***Future development directions for the approach:***

* Research on forgetting risk: Increased research to effectively address the issue of forgetting risk.
* Combination with Explainable AI: Combining Continual Learning and Test Production with Explainable AI to create models that humans can understand and trust more.

**Conclusion**

Combining Continual Learning and Test Production is a powerful approach to building and maintaining effective machine learning models. The effectiveness of this approach has been demonstrated through its ability to maintain flexibility and accuracy of the model over time and new data. However, there are still challenges to be addressed, and continued research is important to improve the continuous learning ability of machine learning models. The combination of Continual Learning and Test Production not only ensures continuity but also helps to shape the reliability and performance of the model in practice.

1. **References**
2. Parisotto, E., & Salakhutdinov, R. (2017). Neural Map: Structured Memory for Deep Reinforcement Learning.
3. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., ... & Hadsell, R. (2017). Overcoming Catastrophic Forgetting in Neural Networks.
4. Maltoni, D., Lomonaco, V., Culurciello, E., & Verschure, P. F. M. J. (2019). Continuous learning in single-incremental-task scenarios.
5. Gorriz, A., Ramirez-Loaiza, M. E., Camps, D., Martin-Herrero, J., & Camps-Valls, G. (2019). Continual learning: A comparative study on how to defy forgetting in classification tasks.
6. Huang, H., Zhuang, B., Li, X., & Wang, L. (2020). Meta-Learning for Few-Shot Continual Learning.
7. Hidasi, B., & Karatzoglou, A. (2018). Recurrent Neural Networks with Top-k Gains for Session-based Recommendations.
8. Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization.
9. Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research, 12(Jul), 2121-2159.
10. Zeiler, M. D. (2012). ADADELTA: An adaptive learning rate method.
11. Hinton, G., Srivastava, N., & Swersky, K. (2012). Lecture 6a Overview of mini-batch gradient descent.
12. Ruder, S. (2016). An overview of gradient descent optimization algorithms.
13. Bottou, L., Curtis, F. E., & Nocedal, J. (2018). Optimization methods for large-scale machine learning. SIAM Review, 60(2), 223-311.
14. Dozat, T. (2016). Incorporating Nesterov momentum into Adam.
15. Reddi, S. J., Kale, S., & Kumar, S. (2019). On the convergence of Adam and beyond.
16. Loshchilov, I., & Hutter, F. (2017). SGDR: Stochastic gradient descent with warm restarts. In International Conference on Learning Representations.
17. Ruder, S. (2017). An overview of gradient descent optimization algorithms.