

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM  
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG  
KHOA CÔNG NGHỆ THÔNG TIN**



**NGUYỄN HOÀNG LINH - 52100907**

**DỰ ÁN CUỐI KỲ**

**BÁO CÁO CUỐI KỲ  
NHẬP MÔN HỌC MÁY**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM  
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG  
KHOA CÔNG NGHỆ THÔNG TIN**



**NGUYỄN HOÀNG LINH – 52100907**

**DỰ ÁN CUỐI KỲ**

**BÁO CÁO CUỐI KỲ  
NHẬP MÔN HỌC MÁY**

**Người hướng dẫn  
PGS. TS. Lê Anh Cường**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

## LỜI CẢM ƠN

Em xin chân thành cảm ơn thầy Lê Anh Cường đã dạy cho em những kiến thức cần thiết để có thể hoàn thành bài báo cáo này.

*TP. Hồ Chí Minh, ngày 18 tháng 12 năm 2023*

*Tác giả*

*(Ký tên và ghi rõ họ tên)*

*Linh*

*Nguyễn Hoàng Linh*

## MỤC LỤC

<b>DANH MỤC CÁC CHỮ VIẾT TẮT.....</b>	<b>iii</b>
<b>CHƯƠNG 1. CÁC PHƯƠNG PHÁP OPTIMIZER TRONG HUẤN LUYỆN MÔ HÌNH HỌC MÁY.....</b>	<b>1</b>
1.1 Giới thiệu.....	1
1.2 Tổng quan về các phương pháp optimizer phổ biến .....	1
1.2.1 <i>Gradient Descent</i> .....	1
1.2.2 <i>Stochastic Gradient Descent (SGD)</i> .....	2
1.2.3 <i>Phương pháp Adaptive Learning Rate</i> .....	4
<b>CHƯƠNG 2. CONTINUAL LEARNING VÀ TEST PRODUCTION KHI XÂY DỰNG DỰNG GIẢI PHÁP HỌC MÁY .....</b>	<b>5</b>
2.1 Giới thiệu về continual learning và test production .....	5
2.1.1 <i>Định nghĩa và tầm quan trọng</i> .....	5
2.1.2 <i>Liên hệ với ngày nay</i> .....	5
2.2 Những thách thức trong continual learning và test production.....	5
2.3 Phương pháp continual learning .....	6
2.4 Các phương pháp thực hành tốt nhất (best practices) trong test production.....	6
2.5 Ứng dụng thực tiễn.....	7
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>8</b>

## **DANH MỤC CÁC CHỮ VIẾT TẮT**

SGD

Stochastic Gradient Descent

# CHƯƠNG 1. CÁC PHƯƠNG PHÁP OPTIMIZER TRONG HUẤN LUYỆN MÔ HÌNH HỌC MÁY

## 1.1 Giới thiệu

Trong lĩnh vực học máy, optimizers đóng vai trò then chốt. Chúng là các thuật toán hoặc phương pháp có nhiệm vụ sửa đổi các thuộc tính của mô hình học máy, chẳng hạn như trọng số, để giảm thiểu các hàm mất mát (loss function). Quá trình này là không thể thiếu trong quá trình đào tạo mô hình vì nó quyết định hiệu quả của việc học. Việc lựa chọn một optimizer thích hợp là rất quan trọng vì nó có thể tác động đáng kể đến khả năng học hỏi và khái quát hóa từ dữ liệu.

## 1.2 Tổng quan về các phương pháp optimizer phổ biến

### 1.2.1 *Gradient Descent*

Gradient Descent là một thuật toán tối ưu hóa thường được sử dụng để đào tạo các mô hình học máy và neural networks. Nó hoạt động bằng cách điều chỉnh lặp đi lặp lại các tham số mô hình để giảm thiểu hàm mất mát (loss function). Thuật toán tính toán độ dốc (gradient) của hàm mất mát (loss function) đối với các tham số và di chuyển theo hướng độ dốc giảm dần (descending gradient). Mặc dù hiệu quả nhưng nhược điểm chính của nó là cần có tốc độ học cố định và có nguy cơ bị mắc kẹt ở local minima trong landscape phức tạp.

#### 1.2.1.1 Tổng quát về cách Gradient Descent hoạt động

Mục tiêu của phương pháp này là giảm thiểu hàm chi phí (cost function), hàm này đo lường sự khác biệt giữa output dự đoán và output thực tế.

**Khởi tạo:** Bắt đầu với các giá trị tham số ban đầu.

**Tính toán độ dốc:** Tính toán độ dốc (bằng đạo hàm riêng) của hàm chi phí (cost function) đối với từng tham số.

**Cập nhật các tham số:** Điều chỉnh các thông số theo hướng ngược lại với gradient.

**Lặp lại:** Lặp lại bước 2 và 3 cho đến khi convergence hoặc đạt số lần lặp được xác định trước.

#### 1.2.1.2 Một số hạn chế của Gradient Descent

Tốc độ học cố định (Fixed Learning Rate): Thuật toán yêu cầu tốc độ học được xác định trước để kiểm soát step size được thực hiện trong mỗi lần lặp. Việc lựa chọn tốc độ học thích hợp có thể là một thách thức, vì tốc độ quá lớn có thể dẫn đến vượt quá mức tối thiểu, trong khi tốc độ quá nhỏ có thể làm chậm quá trình convergence.

Local Minima: Trong các loss landscape phức tạp, Gradient Descent có thể bị mắc kẹt trong local minima, là những điểm thấp hơn khu vực xung quanh nhưng không phải là global minimum. Điều này có thể ngăn cản thuật toán tìm ra giải pháp tối ưu.

#### 1.2.1.3 Tối ưu hóa thuật toán

Lựa chọn tỷ lệ học tập (Learning Rate Choice): Việc tìm kiếm tỷ lệ học tập tối ưu rất quan trọng. Tốc độ cố định có thể phức tạp, dẫn đến convergence chậm hoặc overshooting. Các thuật toán adaptive learning rate như AdaGrad và Adam điều chỉnh tốc độ dựa trên các cập nhật trước đây, giúp quá trình diễn ra suôn sẻ hơn.

Vấn đề khởi tạo: Bắt đầu gần hơn với mức tối thiểu thực tế giúp độ dốc giảm dần convergence nhanh hơn. Việc chọn các giá trị ban đầu thông minh dựa trên kiến thức về domain hoặc đào tạo trước có thể cải thiện đáng kể hiệu quả.

### 1.2.2 Stochastic Gradient Descent (SGD)

Stochastic Gradient Descent, một biến thể của Gradient Descent, cập nhật các tham số mô hình bằng cách sử dụng một training example tại một thời điểm. Bản chất ngẫu nhiên này giúp thoát khỏi local minima, làm cho nó hiệu quả hơn đối với các mô hình phức tạp như deep neural networks. Tuy nhiên, tính ngẫu nhiên của nó có thể dẫn đến sự biến động trên đường đến convergence.

Trong SGD, thay vì sử dụng toàn bộ tập dữ liệu cho mỗi lần lặp, chỉ một mẫu huấn luyện ngẫu nhiên duy nhất (hoặc một small batch) được chọn để tính độ dốc và

cập nhật các tham số mô hình. Lựa chọn này đưa tính ngẫu nhiên vào quá trình tối ưu hóa, do đó có thuật ngữ “Stochastic” trong Stochastic Gradient Descent.

Ưu điểm của việc sử dụng SGD là hiệu quả tính toán, đặc biệt khi xử lý các tập dữ liệu lớn. Bằng cách sử dụng một example duy nhất hoặc một small batch, chi phí tính toán cho mỗi lần lặp lại giảm đáng kể so với các phương pháp Gradient Descent truyền thống yêu cầu xử lý toàn bộ tập dữ liệu.

#### 1.2.2.1 Tổng quan cách Stochastic Gradient Descent hoạt động

**Khởi tạo:** Khởi tạo ngẫu nhiên các tham số của mô hình.

**Thiết lập tham số:** Xác định số lần lặp và tốc độ học (alpha) để cập nhật tham số.

**Vòng lặp Stochastic Gradient Descent:**

Lặp lại các bước sau cho đến khi convergence hoặc đạt số lần lặp tối đa:

1. Xáo trộn tập dữ liệu huấn luyện để tạo ra tính ngẫu nhiên.
2. Lặp lại từng training example (hoặc một small batch) theo thứ tự được xáo trộn.
3. Tính toán độ dốc của hàm chi phí (cost function) đối với các tham số mô hình bằng cách sử dụng example (hoặc batch) hiện tại.
4. Cập nhật các tham số mô hình bằng cách thực hiện một bước theo hướng dốc âm, được chia tỷ lệ theo tốc độ học.
5. Đánh giá các tiêu chí convergence.

Trả về các thông số được tối ưu hóa: Khi tiêu chí hội tụ được đáp ứng hoặc đạt đến số lần lặp tối đa, hãy trả về các tham số mô hình được tối ưu hóa.

#### 1.2.2.2 Ưu và nhược điểm của Stochastic Gradient Descent

**Ưu điểm:**

**Tốc độ:** SGD nhanh hơn các biến thể khác của Gradient Descent.

**Hiệu quả bộ nhớ:** Vì SGD cập nhật từng tham số cho từng training example nên nó tiết kiệm bộ nhớ và có thể xử lý các tập dữ liệu lớn.



Tránh local minima: Do các noisy updates trong SGD, nó có khả năng thoát khỏi local minima và converge đến global minimum..

#### **Nhược điểm:**

Noisy updates: Các update trong SGD rất noisy và có phương sai cao, điều này có thể làm cho quá trình tối ưu hóa kém ổn định hơn và dẫn đến dao động quanh mức tối thiểu.

Convergence chậm: SGD có thể yêu cầu nhiều lần lặp hơn để hội tụ đến mức tối thiểu vì nó cập nhật từng tham số cho từng training example tại một thời điểm.

Độ nhạy đối với tốc độ học: Việc lựa chọn tốc độ học có thể rất quan trọng trong SGD vì việc sử dụng tốc độ học cao có thể khiến thuật toán vượt quá mức tối thiểu, trong khi tốc độ học thấp có thể khiến thuật toán converge chậm.

Kém chính xác hơn: Do noisy updates, SGD có thể không converge đến mức global minimum chính xác và có thể dẫn đến kết quả ở dưới mức tối ưu. Điều này có thể được giảm thiểu bằng cách sử dụng các kỹ thuật như lập kế hoạch tốc độ học tập (learning rate) và cập nhật dựa trên động lượng (momentum-based).

### ***1.2.3 Phương pháp Adaptive Learning Rate***

Phương pháp Adaptive Learning Rate là một công cụ mạnh mẽ trong học máy, được sử dụng để tối ưu hóa các mô hình bằng cách tự động điều chỉnh tốc độ học trong quá trình đào tạo. Chúng giải quyết những hạn chế của tốc độ học cố định, có thể quá chậm hoặc quá nhanh, dẫn đến sự hội tụ (convergence) hoặc phân kỳ (divergence) chậm.

Phương pháp Adaptive Learning Rate là một thuật toán tối ưu hóa điều chỉnh tốc độ học cho từng trọng số trong một mô hình riêng lẻ, dựa trên độ dốc mà nó nhận được. Điều này có nghĩa là các trọng số nhận được độ dốc cao sẽ giảm tốc độ học hiệu quả, trong khi các trọng số nhận được cập nhật nhỏ hoặc không thường xuyên sẽ có tốc độ học hiệu quả tăng lên. Sự điều chỉnh năng động về tỷ lệ học tập này là điều làm cho phương pháp này có tính adaptive. Một trong những ưu điểm chính của phương pháp này là khả năng xử lý dữ liệu thưa thớt và độ dốc nhiễu (noisy gradient).

## **CHƯƠNG 2. CONTINUAL LEARNING VÀ TEST PRODUCTION KHI XÂY DỰNG DỰNG GIẢI PHÁP HỌC MÁY**

Continual learning, hay còn được gọi là học tập suốt đời và test production là những khía cạnh quan trọng trong việc phát triển và duy trì các giải pháp học máy hiệu quả.

### **2.1 Giới thiệu về continual learning và test production**

#### ***2.1.1 Định nghĩa và tầm quan trọng***

Continual đề cập đến khả năng của một mô hình học máy có thể học liên tục, thích ứng với dữ liệu mới trong khi vẫn giữ được knowledge đã thu được trước đó. Khái niệm này rất quan trọng trong môi trường dynamic, nơi việc phân phối dữ liệu thay đổi theo thời gian, đòi hỏi các mô hình phải phát triển mà không quên những knowledge đã học trước đó.

Mặt khác, test production bao gồm quá trình thử nghiệm nghiêm ngặt các mô hình máy học trong môi trường sản xuất để đảm bảo chúng hoạt động như mong đợi trên dữ liệu thế giới thực. Điều quan trọng là xác nhận tính hiệu quả và độ tin cậy của mô hình trước và sau khi triển khai.

#### ***2.1.2 Liên hệ với ngày nay***

Trong thời đại mà dữ liệu không ngừng phát triển, khả năng thích ứng và hoạt động một cách đáng tin cậy của các mô hình trong sản xuất là điều rất quan trọng và cần thiết. Khả năng thích ứng này đảm bảo rằng các giải pháp máy học vẫn hoạt động hiệu quả theo thời gian, giải quyết các vấn đề như sai lệch khái niệm (concept drift) và model decay.

### **2.2 Những thách thức trong continual learning và test production**

**Forgetting problem:** Một trong những thách thức chính trong việc continual learning là vấn đề quên (forgetting problem), trong đó các mô hình mất đi knowledge về các task trước đó khi dữ liệu mới được đưa vào. Vấn đề này là một trở ngại đáng kể trong việc đạt được khả năng học tập suốt đời (lifelong learning).

**Data Distribution và Quality:** Việc đảm bảo chất lượng và tính đại diện của dữ liệu trong việc thiết lập continual learning là một việc khó khăn. Việc phân phối dữ liệu có thể thay đổi theo thời gian, có khả năng dẫn đến sai lệch hoặc không chính xác của mô hình.

**Sự phức tạp trong test production:** Các mô hình thử nghiệm trong môi trường sản xuất rất phức tạp do tính không thể đoán trước và tính biến đổi của dữ liệu trong thế giới thực. Đảm bảo rằng các mô hình hoạt động tốt trong nhiều tình huống khác nhau là một nhiệm vụ không hề đơn giản.

## 2.3 Phương pháp continual learning

**Regularization Techniques:** Regularization Techniques như Elastic Weight Consolidation (EWC) giúp giảm thiểu vấn đề quên bằng cách xử phạt (penalize) các thay đổi đối với các trọng số quan trọng của mô hình vốn rất quan trọng đối với các task trước đó.

**Dynamic Architectures:** Các kiến trúc thần kinh động (dynamic neural architectures), chẳng hạn như Progressive Neural Networks, cho phép mô hình mở rộng với các task mới, cung cấp cách giữ lại knowledge cũ trong khi thu thập thông tin mới.

**Experience Replay:** Experience replay bao gồm việc lưu trữ một tập hợp con các dữ liệu trước đó và đào tạo lại mô hình trên dữ liệu đó một cách định kỳ, cùng với dữ liệu mới, để duy trì knowledge theo thời gian.

## 2.4 Các phương pháp thực hành tốt nhất (best practices) trong test production

**Giám sát và đánh giá liên tục:** Việc giám sát liên tục hiệu suất của mô hình trong sản xuất là điều cần thiết. Bao gồm việc thiết lập các metrics và cảnh báo để

phát hiện tình trạng suy giảm hiệu suất (performance degradation) hoặc trôi dạt dữ liệu (data drift).

**A/B Testing:** A/B testing bao gồm việc so sánh hiệu suất của các phiên bản mô hình mới với mô hình sản xuất hiện tại để đảm bảo những cải tiến trước khi triển khai trên quy mô lớn.

**Kiểm tra độ bền (robustness testing):** Kiểm tra độ bền, bao gồm kiểm tra sức chịu đựng và phân tích kịch bản (scenario analysis), giúp đánh giá cách mô hình hoạt động trong các điều kiện khắc nghiệt hoặc bất thường.

## 2.5 Ứng dụng thực tiễn

Continual learning và test production rất quan trọng trong các ứng dụng như xe tự hành, chẩn đoán chăm sóc sức khỏe và dự báo tài chính, nơi các mô hình phải thích ứng với dữ liệu và môi trường mới.

Khi các mô hình liên tục học hỏi và phát triển, việc giải quyết các cân nhắc về đạo đức như quyền riêng tư, công bằng và minh bạch ngày càng trở nên quan trọng.

Những tiến bộ trong thuật toán, phần cứng và thực hành xử lý dữ liệu có khả năng giúp continual learning và test production trở nên hiệu quả hơn, mở đường cho các giải pháp học máy mạnh mẽ và thích ứng (adaptive) hơn.

## TÀI LIỆU THAM KHẢO

Tiếng Anh

What is Gradient Descent? | IBM. (n.d.).

<https://www.ibm.com/topics/gradient-descent#:~:text=Gradient%20descent%20is%20an%20optimization,each%20iteration%20of%20parameter%20updates.>

Gupta, A. (2023, September 13). A comprehensive guide on optimizers in deep learning. Analytics Vidhya. <https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on-deep-learning-optimizers/>

GeeksforGeeks. (2023, May 29). ML Stochastic Gradient Descent SGD. <https://www.geeksforgeeks.org/ml-stochastic-gradient-descent-sgd/>

Wang, L. (2023, January 31). A Comprehensive survey of Continual Learning: Theory, Method and application. arXiv.org. <https://arxiv.org/abs/2302.00487>

Introduction to Continual Learning - Wiki. (n.d.). <https://wiki.continualai.org/the-continualai-wiki/introduction-to-continual-learning>

Barla, N. (2023, July 31). 5 tools that will help you setup Production ML Model testing. neptune.ai. <https://neptune.ai/blog/tools-ml-model-testing>