TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**MAI NHẬT DUY - 52100882**

**TÌM HIỂU LÝ THUYẾT HỌC MÁY**

**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**



**MAI NHẬT DUY - 52100882**

**TÌM HIỂU LÝ THUYẾT HỌC MÁY**

**BÁO CÁO CUỐI KỲ**

**NHẬP MÔN  
HỌC MÁY**

Người hướng dẫn

**TS. LÊ ANH CƯỜNG**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

**LỜI CẢM ƠN**

Chúng em xin chân thành cảm ơn đến Trường Đại học Tôn Đức Thắng đã mở môn học Học Máy. Sự đầu tư và tâm huyết của trường trong việc giáo dục và đào tạo đã tạo ra một môi trường học tập tuyệt vời, giúp sinh viên như em có cơ hội tiếp cận và nắm bắt kiến thức mới.

Đặc biệt, em muốn gửi lời cảm ơn sâu sắc đến Thầy Lê Anh Cường. Sự tận tâm và nhiệt huyết của thầy trong việc truyền đạt kiến thức đã tạo nên những bài giảng sinh động và thú vị. Thầy luôn quan tâm đến sinh viên, tạo điều kiện thuận lợi để chúng em có thể học hỏi và phát triển.

*TP. Hồ Chí Minh, ngày ... tháng … năm 2023*

*Tác giả*

*(Ký tên và ghi rõ họ tên)*

Duy

Mai Nhật Duy

**CÔNG TRÌNH ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi và được sự hướng dẫn khoa học của TS.Lê Anh Cường. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong Dự án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung Dự án của mình**. Trường Đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày … tháng … năm 2023*

*Tác giả*

*(Ký tên và ghi rõ họ tên)*

*Duy*

*Mai Nhật Duy*

**TÊN ĐỀ TÀI**

**TÓM TẮT**

Tìm nhiều về các thuật toán Optimizer.

Tìm hiểu về Continual Learning và Test Production.

**TITLE**

**ABSTRACT**

**MỤC LỤC**

[DANH MỤC HÌNH VẼ vii](#_Toc154231174)

[DANH MỤC BẢNG BIỂU viii](#_Toc154231175)

[DANH MỤC CÁC CHỮ VIẾT TẮT ix](#_Toc154231176)

[CHƯƠNG 1. PHƯƠNG PHÁP OPTIMIZER TRONG HUẤN LUYỆN MÔ HÌNH HỌC MÁY. 1](#_Toc154231177)

[1.1 Mục tiêu của các phương pháp Optimizer 1](#_Toc154231178)

[1.2 Các thuật toán tối ưu 1](#_Toc154231179)

[1.2.1 Gradient Descent (GD) 1](#_Toc154231180)

[1.2.2 Momentum 5](#_Toc154231181)

[1.2.3 Adagrad 6](#_Toc154231182)

[1.2.4 RMSprop 7](#_Toc154231183)

[1.2.5 AdaDelta 8](#_Toc154231184)

[1.2.6 Adam 9](#_Toc154231185)

[1.3 Tổng kết về optimizer 10](#_Toc154231186)

[CHƯƠNG 2. CONTINUAL LEARNING VÀ TEST PRODUCTION 12](#_Toc154231187)

[2.1 Continual Learning 12](#_Toc154231188)

[2.1.1 Giới thiệu Continual Learning 12](#_Toc154231189)

[2.1.2 Các đặc trưng của Continual Learning 13](#_Toc154231190)

[2.2 Test Production 13](#_Toc154231191)

[2.2.1 Khái niệm 13](#_Toc154231192)

[2.2.2 Các phương pháp kiểm tra 15](#_Toc154231193)

[CHƯƠNG 3. KẾT LUẬN 15](#_Toc154231194)

[TÀI LIỆU THAM KHẢO 17](#_Toc154231195)

# DANH MỤC HÌNH VẼ

[Hình 1.1: Mô tả Gradient Descent 1](#_Toc142677545)

[Hình 1.2: Các bước duyệt dữ liệu của Stichastic Gradient Descent. 3](#_Toc142677545)

[Hình 1.3: Các bước duyệt dữ liệu của Mini-Batch Gradient Descent 4](#_Toc142677545)

[Hình 1.4: Mô tả thuật toán momenton 5](#_Toc142677545)

[Hình 1.5: Các bước duyệt dữ liệu của Adagrad 6](#_Toc142677545)

[Hình 1.6: Các bước duyệt dữ liệu của RMSprop 7](#_Toc142677545)

[Hình 1.7: Hiệu suất của các giải thuật optimizer 10](#_Toc142677545)

# DANH MỤC BẢNG BIỂU

# DANH MỤC CÁC CHỮ VIẾT TẮT

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

## Mục tiêu của các phương pháp Optimizer

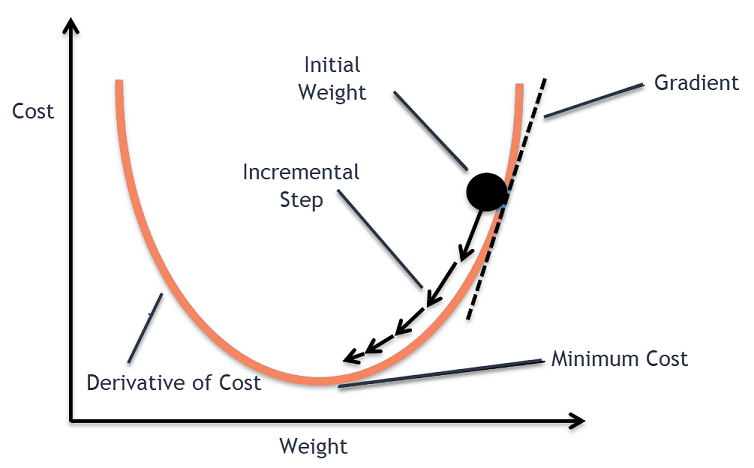
Trong học máy, một bước quan trọng trong quá trình huấn luyện mô hình là tối ưu hóa. Optimizer (Bộ tối ưu hóa) là một thành phần quan trọng trong học máy và được sử dụng để điều chỉnh các tham số của mô hình dự đoán dựa trên dữ liệu huấn luyện.

Mục tiêu của tối ưu hóa là tìm ra các giá trị tham số tối ưu (weights, bias) nhằm giảm thiểu sai số (đánh giá) giữa kết quả dự đoán và giá trị thực tế trong quá trình huấn luyện. Tuy nhiên, việc tìm ra các giá trị tối ưu này thường rất phức tạp với các mô hình học máy phức tạp. Vì thế chúng ta phải tìm 1 thuật toán để cải thiện weight và bias theo từng bước, và đó là lý do các thuật toán optimizer ra đời.

## Các thuật toán tối ưu

### Gradient Descent

Giới thiệu cơ bản về Gradient Descent:

Hình 1.1: Mô tả Gradient Descent

Gradient Descent là một thuật toán tìm tối ưu chung cho các hàm số. Ý tưởng chung của GD là điều chỉnh các tham số để lặp đi lặp lại thông qua mỗi dữ liệu huấn luyện để giảm thiểu hàm chi phí.

Trong học máy, hàm chi phí thường là hàm mất mát (loss function), đại diện cho sự sai lệch giữa kết quả dự đoán của mô hình và dữ liệu thực tế.

* + là tập hợp các tham số mới của loss function sau khi đạo hàm
  + là tập hợp các tham số hiện tại của hàm loss function
  + là learning rate
  + là đạo hàm của hàm loss function tại .

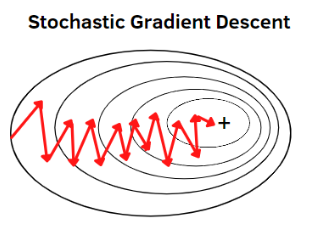
Lưu ý: **Luôn luôn đi ngược hướng với đạo hàm**.

Gradient Descent là thuật toán quan trọng, là nền tảng để phát triển lên nhiều thuật toán có độ hiệu quả cao. Gradient Descent có nhiều biến thể để tối ưu hóa việc ra giá trị cực tiểu của loss function.

Đánh giá:

* + Ưu điểm: Dễ triển khai, hiệu quả đối với hàm số lồi, nền tảng để phát trển các thuật toán tối ưu khác.
  + Nhược điểm: Phức tạp do yêu cầu tính toán trên toàn tập huấn huyện tốc độ xử lý chậm, hoạt động không tốt với bộ dữ liệu lớn, không linh hoạt, phụ thuộc vào vị trí khởi tạo mà có thể cho ra kết quả khác nhau do bị rơi vào cực tiểu cục bộ, bị ảnh hưởng bởi learning rate, không hiệu quả với oline lerning.
  + Khắc phục: Để khắc phục và cải tiến Gradient Descent thì các biến thể Gradient Descent đã được ra đời như Stochastic Gradient Descent,Mini-batch Gradient Descent,...

#### Stochastic Gradient Descent



Hình 1.2: các bước duyệt dữ liệu của Stichastic Gradient Descent.

Lý thuyết: Trong thuật toán này, tại 1 thời điểm, ta chỉ tính đạo hàm của hàm mất mát dựa trên chỉ một điểm dữ liệu  với thứ tự chọn điểm dữ liệu là ngẫu nhiên rồi cập nhật  dựa trên đạo hàm này. Việc này được thực hiện với từng điểm trên toàn bộ dữ liệu, sau đó lặp lại quá trình trên.

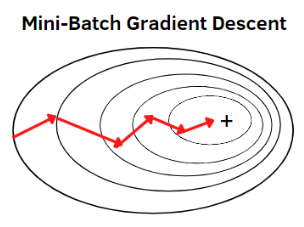
* + là tập hợp các tham số mới của loss function sau khi đạo hàm
  + là tập hợp các tham số hiện tại của loss function
  + là learning rate
  + là đạo hàm của loss function chỉ với một điểm dữ liệu huấn luyện.

Đánh giá:

* Ưu điểm: tốc độ nhanh và cho ra các tham số chấp nhận được, thích hợp sử dụng cho bộ dữ liệu lớn, có thể thoát khỏi cực tiểu địa phương,
* Nhược điểm: không ổn định do cách chọn điểm dữ liệu là ngẫu nhiên, độ dao động cao nên gặp khó khăng khi muốn tìm ra cực tiểu tối ưu, không phù hợp ứng dụng cho bộ dữ liệu cập nhật liên tục(online learning).

Khắc phục: bằng cách giảm dần learning rate khi bắt đầu chạy ta sẽ đặt learning rate lớn để thoát khỏi cực tiểu địa phương sau đó learning rate sẽ được giảm dần để cố đạt được cực tiểu toàn cục. Đồng thời lưu ý nếu tốc độ giảm quá nhanh thì có thể sẽ vướng vào cực tiểu địa phương, còn nếu giảm quá chậm thì lại tốn thời gian.

#### Mini-Batch Gradient Descent



Hình 1.3: Các bước duyệt dữ liệu của Mini-Batch Gradient Descent.

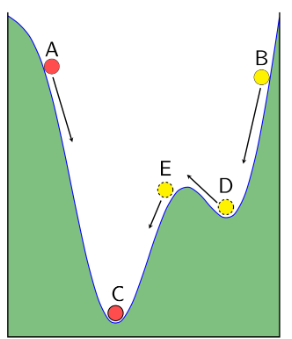
Lý thuyết: Thay vì tính toán vector độ dốc dựa trên toàn bộ tập huấn luyện (GD) hoặc dựa trên 1 điểm dữ liệu (Stochastic GD) thì Mini-batch GD tính gradients dựa trên 1 tập nhỏ ngẫu nhiên các điểm dữ liệu được gọi là mini-batches.

* + là dữ liệu từ đến thứ
  + là tập hợp các tham số mới của loss function sau khi đạo hàm
  + là tập hợp các tham số hiện tại của loss function
  + là learning rate
  + là đạo hàm của loss function chỉ với một điểm dữ liệu huấn luyện.

Đánh giá:

* Ưu điểm: Tốc độ nhanh, ổn định, linh hoạt trong việc kết hợp các hàm optimizer khác, có thể ứng dụng cho online learning.
* Nhược điêm: Độ dao động cao đòi hỏi phải điều chỉnh kích thước mini-batch và tốc độ học (learning rate) một cách cẩn thận để được kết quả tối ưu.

### Momentum



Hình 1.4: Mô tả thuật toán momenton.

Lý thuyết: Dựa trên hiện tượng vật lý khi thả hai quả bóng xuống thung lũng có hai vùng trũng là D và C thì cả hai đều có vị trí cuối cùng tại C, có được kết quả như vậy là nhớ có quán tính và tốc độ lăn của quả bóng. Để biểu diễn hiện tượng này bằng toán học ta có công thức như sau.

Ta xem là vận tốc và nó vừa phải mang giá trị vừa mang giá trị độ dốc .

Sau đó vị trí mới của quả bóng sẽ là

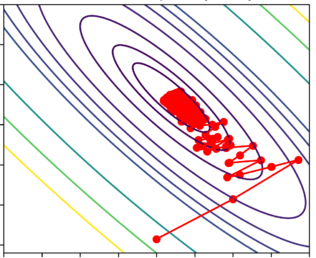
* + hệ số momentum thường có giá trị là
  + là tập hợp các tham số hiện tại của loss function
  + là learning rate
  + là độ dốc của điểm trước đó.

Đánh giá:

Ưu điểm: điểm sáng nhất của momentum là việc nó có khả năng thoát ra khỏi cực tiểu địa phương và tìm đến cực tiểu toàn cục, nhờ việc tích lũy thông tin từ vị trí trức đó mà tốc độ hội tụ của momentum được cải thiện hơn, đồng thời độ dao động cũng không cao.

Nhược điểm: dễ bị ảnh hưởng bởi tham số, và phải chú ý hiệu chỉnh hệ số momentum phù hợp.

### Adagrad



Hình 1.5: Các bước duyệt dữ liệu của Adagrad.

Lý thuyết: Ý tưởng của adagrad là cập nhật lại giá trị learning rate mới từ các giá trị gradiant cũ.

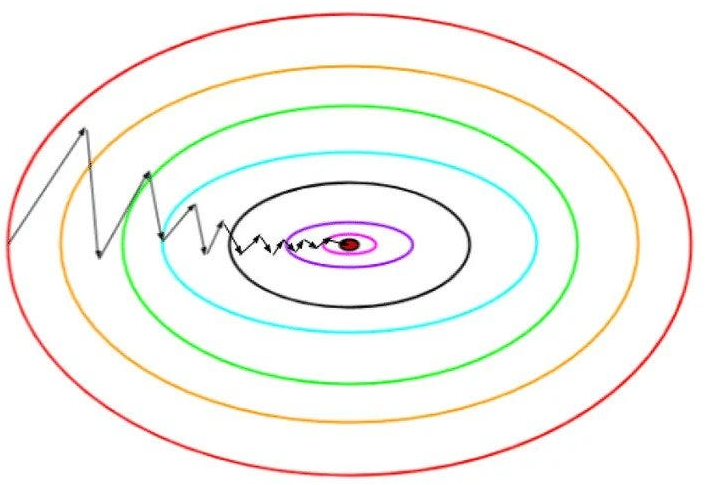
Mà:

* + learning rate mới được cập nhật từ dữ liệu cũ.
  + learning rate cũ.
  + giá trị sắp xĩ bằng 0 để tránh mẫu số bằng 0
  + là tổng bình phương các gradient được tính trước đó .
  + là độ dốc tại thời điểm t

Đánh giá:

* Ưu điểm: không cần điều chỉnh thủ công learning rate, tốc độ cao, kết quả đáng tin cậy, có thể ứng dụng chó online learning.
* Nhược điểm: trong quá trình lặp thì giá trị có thể trở nên lớn khiến cho learning rate nhỏ dẫn đến quá trình tìm cực tiểu bị chậm.

### RMSprop



Hình 1.6: Các bước duyệt dữ liệu của RMSprop.

Lý thuyết: xây dựng dựa trên Adagrad, nhưng RMSprop như một bản vá đơn giản để tách rời tốc độ định thời ra khỏi tốc độ học thay đổi theo tọa độ đã khắc phục việc giá trị tăng không kiểm soát làm cho tốc độ thực thi giảm bằng cách sử dụng trung bình rò rỉ tương tự như trong phương pháp động lượng để xác dịnh số các giá cũ được dùng để tạo nên learning rate mới. giá trị được cải tiến như sau.

Mà:

* + learning rate mới được cập nhật từ dữ liệu cũ.
  + learning rate cũ.
  + giá trị sắp xĩ bằng 0 để tránh mẫu số bằng 0
  + hệ số xác định độ dài thông tin quá khứ được sử dụng để điểu chình tỉ lệ học.
  + là tổng bình phương gradient được tính trước đó .
  + là độc dốc tại thời điểm t

Đánh giá:

Ưu điểm: RMSprop có các ưu điểm của Adagrad nhưng cải tiến hơn khi có tham số điều chỉnh được số lượng giá trị trong quá khứ được sử dụng để tạo nên learning rate mới.

Nhược điểm: cần chọn tham số hợp lý tránh trường hợp không thoát ra khỏi cực tiểu cục bộ hoặc tộc độ chạy chậm.

### AdaDelta

Lý thuyết:

Adadelta sử dụng hai biến trạng thái,  để lưu trữ trung bình rò rỉ mô-men bậc hai của gradient và  để lưu trữ trung bình rò rỉ mô-men bậc hai của lượng thay đổi của các tham số trong mô hình

Trong đó bản cập nhật tham số được thay đổi như sau:

* + learning rate mới được cập nhật từ dữ liệu cũ.
  + l.
  + giá trị sắp xĩ bằng 0 để tránh mẫu số bằng 0.
  + là tham số suy giảm thường là 0.9.
  + là độ dốc tại t.

Đánh giá:

Ưu điểm: làm tăng tốc độ bằng cách sử dụng tốc độ thay đổi của chính bản thân các tham số để điều chỉnh tốc độ học nhưng vẫn đảm bảo tính chính xác như Adagrad.

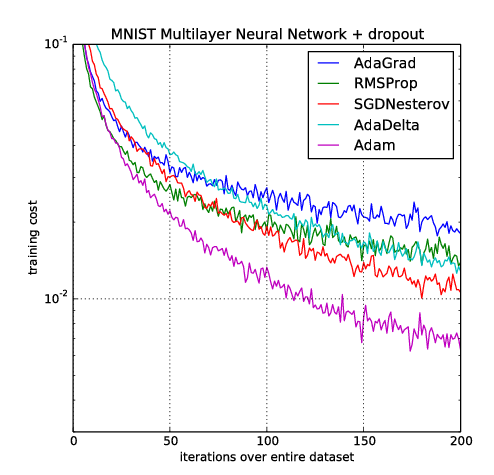
Nhược điểm: AdaDelta cần lưu trữ thông tin về mức độ thay đổi của hàm chi phí trong quá trình học tập. Điều này có thể yêu cầu nhiều bộ nhớ, đặc biệt nếu tập dữ liệu lớn, Có thể bị kẹt trong điểm cực tiểu cục bộ.

### Adam

Lý thuyết: **Adam** kết hợp các thuộc tính tốt nhất của các thuật toán AdaGrad và RMSP để cung cấp một thuật toán tối ưu hóa có thể xử lý được các gradient thưa thớt trên các bài toán nhiễu. Adam tính toán trung bình động mũ của gradient và bình phương gradient. Và các tham số β1​ và β2​ được sử dụng để điều khiển tốc độ suy giảm của các trung bình động này. Ban đầu, cả ​ và ​ được đặt bằng 0. Cả hai đều có xu hướng chệch về 0 khi β1​ và β2​ bằng 1. Bằng cách tính toán ​ và làm giảm thiểu sai lệch, vấn đề này được Adam khắc phục.

* ​ và ​ là trung bình động mũ của moment bậc nhất và bậc hai của gradient .
* và ​ là các giá trị được hiệu chỉnh để giảm thiểu sai lệch,
* α là tốc độ học,
* ϵ là một số rất nhỏ để tránh chia cho 0
* là trọng số tại thời điểm t

Đánh giá:



Hình 1.7 Hiệu suất của các giải thuật optimizer

Ưu điểm: Hiệu suất thử nghiệm vượt trội hơn so với các optimizer khác nhất là khi được ứng dụng cho bộ dữ liệu lớn, dễ triển khai, tốn ít bộ nhớ

Nhược điểm: Vẫn như các giải thuật optimizer khác Adam vẫ có thể bị kẹt ở cực tiểu cục bộ.

## Tổng kết về optimizer

Có nhiều thuật toán optimizer đã được phát triển nhằm cải thiện mô hình học máy. Mỗi thuật toán có ưu và nhược điểm riêng của mình vì thể lập trình viên có thể chọn thuật toán có thể đáp ứng yêu cầu của bài toán.

# CONTINUAL LEARNING VÀ TEST PRODUCTION

## Continual Learning

### Giới thiệu Continual Learning

Học liên tục (Continual Learning) hay (Lifelong learning), được xây dựng dựa trên ý tưởng về việc học tập liên tục về thế giới bên ngoài để cho phép phát triển tự động, tăng cường trao dồi kỹ năng và kiến thức phức tạp hơn. Một mô hình học máy giờ không chỉ có thể thực hiện một yêu cầu một nhiệm vụ nữa mà còn có thể thực hiện nhiều nhiệm vụ khác nhau.

Khi mô hình học liên tục được huấn luyện trong một môi trường phức tạp, nó có thể tìm ra các chiến lược, hành vi hoặc kết quả mới mà không được cung cấp trước đó. Mô hình có thể khám phá và tìm ra các hành động mới hoặc tập hợp các hành động đã biết để tạo ra kết quả tốt hơn.

Một hệ thống học liên tục có thể được định nghĩa là một thuật toán thích ứng có khả năng học từ một dòng thông tin liên tục, với thông tin này dần dần có sẵn theo thời gian và nơi số lượng nhiệm vụ cần học (ví dụ: các lớp thành viên trong một nhiệm vụ phân loại) không được xác định trước. Điều quan trọng là việc tiếp nhận thông tin mới nên diễn ra mà không gây ra quên lãng hoặc can thiệp thảm họa.

Do đó, trong tình huống Học liên tục, một mô hình học cần phải xây dựng và cập nhật liên tục các biểu diễn nội tại khi phân phối của các nhiệm vụ thay đổi một cách động trong suốt quãng đời của nó. Lý tưởng nhất, một phần của các biểu diễn nội tại như vậy sẽ chung chung và bất biến đủ để có thể tái sử dụng trên các nhiệm vụ tương tự, trong khi một phần khác nên bảo tồn và mã hóa các biểu diễn cụ thể cho nhiệm vụ.

Một trong những vấn đề mở quan trọng và đầy thách thức nhất trong Trí tuệ nhân tạo (AI) là vấn đề học liên tục. Một mô hình học máy học liên tục xem xét các hệ thống có thể liên tục học nhiều tác vụ (từ một hoặc nhiều miền) trong suốt cuộc đời. Một hệ thống học liên tục hiệu quả và hiệu quả:

1. giữ lại kiến thức đã học được từ các nhiệm vụ khác nhau;
2. chuyển giao có chọn lọc kiến thức (từ các nhiệm vụ đã học trước đó) để tạo điều kiện thuận lợi cho việc học các nhiệm vụ mới;
3. đảm bảo sự tương tác hiệu quả và hiệu quả giữa (1) và (2).

học liên tục giới thiệu một số thách thức cơ bản trong các mô hình đào tạo thường không phát sinh trong một môi trường học tập hàng loạt nhiệm vụ duy nhất. Điều này bao gồm các vấn đề như quên thảm khốc và bão hòa dung lượng. Hội thảo này nhằm mục đích khám phá các giải pháp cho những vấn đề này trong cả môi trường học tập có giám sát và học tăng cường.

### Các đặc trưng của Continual Learning

Học liên tục được đặc trưng trong thực tế bằng một loạt các yêu cầu:

* **Học trực tuyến(online learning)**: việc học diễn ra mọi lúc, không có các nhiệm vụ cố định hay tập dữ liệu cố định và không có ranh giới rõ ràng giữa các nhiệm vụ.
* **Sự chuyển giao (forward/backward)**: người học nên có khả năng chuyển giao và thích ứng những gì đã học từ kinh nghiệm, dữ liệu, hoặc nhiệm vụ trước đó sang các tình huống mới, cũng như sử dụng kinh nghiệm gần đây để cải thiện hiệu suất trên các khả năng đã học trước đó.
* **Kháng quên lãng**: việc học mới không nên phá hủy hiệu suất trên dữ liệu đã nhìn thấy trước đó.
* **Kích thước hệ thống giới hạn**: khả năng học của người học nên được cố định, buộc hệ thống phải sử dụng nguồn lực của mình một cách thông minh, quên đi những gì đã học một cách nhẹ nhàng để giảm thiểu mất mát tiềm năng của phần thưởng tương lai.
* **Không truy cập trực tiếp vào kinh nghiệm trước đó**: trong khi mô hình có thể nhớ một lượng hạn chế kinh nghiệm, một thuật toán học liên tục không thể giả định có quyền truy cập trực tiếp vào tất cả kinh nghiệm quá khứ của nó hoặc khả năng tua lại môi trường (tức là, t=0 chỉ một lần).

## Test Production

### Khái niệm

Test Production trong học máy là quá trình kiểm tra và đánh giá mô hình được huấn luyện khi triển khai vào thực tế. Trong quá trình Test Production, mô hình được chạy trên dữ liệu thực tế hoặc môi trường thực tế để đánh giá khả năng của nó trong việc thực hiện các dự đoán hoặc nhiệm vụ cụ thể.

Mục tiêu của Test Production là đảm bảo rằng mô hình hoạt động đúng đắn và cho hiệu suất tốt trong môi trường thực tế. Quá trình này giúp xác nhận rằng mô hình đã được huấn luyện trên dữ liệu đại diện và có khả năng tổng quát hóa tốt sang dữ liệu mới. Nó cũng giúp phát hiện và khắc phục các sự cố hoặc vấn đề có thể phát sinh khi triển khai mô hình.

* Xác định yêu cầu và mục tiêu của bài toán: Đầu tiên, bạn cần hiểu rõ bài toán mà bạn đang cố gắng giải quyết. Xác định rõ yêu cầu, kết quả dự kiến và các chỉ số đánh giá mà bạn muốn đạt được với giải pháp học máy của mình. Điều này sẽ định hình quá trình Test Production và các bước tiếp theo.
* Xử lý và chuẩn bị dữ liệu: Tiếp theo, bạn cần xử lý và chuẩn bị dữ liệu để sử dụng trong quá trình Test Production. Đảm bảo rằng dữ liệu được đại diện cho môi trường thực tế và bao gồm các trường hợp đại diện cho các tình huống khác nhau mà mô hình có thể gặp phải.
* Triển khai mô hình: Bước tiếp theo là triển khai mô hình học máy vào môi trường thực tế. Điều này có thể bao gồm việc xây dựng một hệ thống hoặc ứng dụng để triển khai mô hình và tích hợp nó với hạ tầng hiện có. Đảm bảo rằng mô hình được triển khai một cách đúng đắn và hoạt động như mong đợi.
* Kiểm tra tính toàn vẹn dữ liệu: Trước khi thực hiện các bước kiểm tra khác, hãy đảm bảo rằng dữ liệu đầu vào cho mô hình là hợp lệ và đúng đắn. Kiểm tra định dạng, giá trị ngoại lai và tính nhất quán của dữ liệu để đảm bảo rằng mô hình sẽ nhận được dữ liệu chính xác và tin cậy.
* Kiểm tra hiệu suất: Chạy mô hình trên dữ liệu thực tế và đánh giá hiệu suất của nó. Đo lường các chỉ số đánh giá như độ chính xác, độ phân loại, độ đo lỗi và thời gian phản hồi để đảm bảo rằng mô hình hoạt động như mong đợi và đáp ứng được yêu cầu và mục tiêu của bài toán.
* Kiểm tra độ tin cậy và bảo mật: Đảm bảo rằng mô hình đáp ứng các yêu cầu về độ tin cậy và bảo mật trong môi trường thực tế. Kiểm tra các biện pháp bảo mật và xác thực được áp dụng, đảm bảo rằng mô hình không có sự rò rỉ thông tin nhạy cảm và tuân thủ các quy tắc quyền riêng tư.
* Theo dõi và cải thiện: Sau khi triển khai mô hình, quá trình Test Production cũng bao gồm việc theo dõi và cải thiện hiệu suất của mô hình trong thời gian thực. Theo dõi quá trình triển khai và thu thập phản hồi từ người dùng và hệ thống để xác định các vấn đề, lỗi và cải thiện tiềm năng. Dựa trên thông tin này, bạn có thể điều chỉnh và cải thiện mô hình để đạt được hiệu suất tốt hơn và đáp ứng được yêu cầu của bài toán.

### Các phương pháp kiểm tra

Có nhiều phương pháp kiểm thử được áp dụng trong quá trình Test Production để đảm bảo tính toàn vẹn và hiệu suất của mô hình học máy trong môi trường thực tế. Dưới đây là một số phương pháp kiểm thử phổ biến:

* Kiểm tra đơn vị (Unit testing): Phương pháp này tập trung vào việc kiểm tra từng phần riêng lẻ của mô hình, chẳng hạn như các hàm, phương thức hoặc thành phần nhỏ. Điều này giúp đảm bảo tính chính xác và đúng đắn của các phần tử cơ bản trong mô hình.
* Kiểm tra tích hợp (Integration testing): Phương pháp này kiểm tra khả năng hoạt động của các thành phần khác nhau trong mô hình khi được kết hợp với nhau. Nó đảm bảo rằng các thành phần hoạt động một cách liên tục và tương tác chính xác với nhau.
* Kiểm tra hồi quy (Regression testing): Phương pháp này được sử dụng để đảm bảo rằng các thay đổi hoặc cải tiến được thực hiện trong mô hình không gây ra các vấn đề mới hoặc ảnh hưởng đến các chức năng hiện có. Nó đảm bảo rằng mô hình vẫn hoạt động như mong đợi sau khi thay đổi được áp dụng.
* Kiểm tra chức năng (Functional testing): Phương pháp này kiểm tra tính chính xác và hiệu suất của mô hình bằng cách kiểm tra các chức năng và khả năng của nó. Điều này đảm bảo rằng mô hình đáp ứng đúng các yêu cầu và mục tiêu chức năng trong bài toán.
* Kiểm tra hiệu suất (Performance testing): Phương pháp này đánh giá hiệu suất của mô hình trong các điều kiện khác nhau, chẳng hạn như tải cao, dữ liệu lớn hoặc thời gian thực. Nó đảm bảo rằng mô hình có thể xử lý và đáp ứng được yêu cầu về hiệu suất trong môi trường thực tế.
* Kiểm tra tự động (Automated testing): Phương pháp này sử dụng các công cụ và kịch bản tự động để thực hiện các bài kiểm tra một cách tự động. Điều này giúp tiết kiệm thời gian và tăng tính nhất quán của quá trình kiểm thử.
* Kiểm tra A/B (A/B testing): Phương pháp này so sánh hiệu suất và tác động của hai phiên bản mô hình khác nhau (A và B) trên cùng một tập dữ liệu hoặc nhóm người dùng. Nó giúp xác định phiên bản nào mang lại kết quả tốt hơn và hướng dẫn việc cải thiện mô hình.

Những phương pháp kiểm thử này có thể được kết hợp và tùy chỉnh phù hợp với bài toán cụ thể và môi trường triển khai. Mục tiêu chung là đảm bảo rằng mô hình hoạt động đúng đắn, đáp ứng yêu cầu và mục tiêu, và có tính toàn vvẹn trong môi trường thực tế.

# KẾT LUẬN

Những nội dung về Optimizer hay Continual Learning và Test Production đều là những yếu tố quang trọng để xây dựng mô hình học máy có hiệu suất và tính ứng dụng thực tế cao, giúp ích cho xã hội.

# TÀI LIỆU THAM KHẢO

Tiếng Việt

[VIBLO](https://viblo.asia/followings)

[Wikipedia, bách khoa toàn thư mở](https://vi.wikipedia.org/wiki/Trang_Ch%C3%ADnh)

Tiếng Anh

[Introduction to Continual Learning - Wiki (continualai.org)](https://wiki.continualai.org/the-continualai-wiki/introduction-to-continual-learning)