

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



**ĐỒ ÁN CUỐI KỲ**

**NHẬP MÔN HỌC MÁY**

*Người hướng dẫn:* **Giảng viên LÊ ANH CƯỜNG**

*Người thực hiện:* **ĐINH PHÚ QUỐC – 52100927**

**Lớp: 21050301**

**Khóa: 25**

**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**



**ĐỒ ÁN CUỐI KỲ**

# **NHẬP MÔN HỌC MÁY**

*Người hướng dẫn:* **Giảng viên LÊ ANH CƯỜNG**

*Người thực hiện:* **ĐINH PHÚ QUỐC – 52100927**

**Lớp: 21050301**

**Khóa: 25**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

## **LỜI CẢM ƠN**

Em xin gửi lời cảm ơn sâu sắc nhất đến thầy Lê Anh Cường vì đã giúp em có những kiến thức cần thiết để hoàn thành báo cáo này.

## **ĐỒ ÁN ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là sản phẩm đồ án của riêng tôi / chúng tôi và được sự hướng dẫn của Giảng viên 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.

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

*TP. Hồ Chí Minh, ngày 23 tháng 12 năm 2023*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Đinh Phú Quốc*

## PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

### 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)

## TÓM TẮT

Báo cáo này là bài đồ án cuối kỳ môn Nhập môn học máy năm 2023, với các nội dung chính như sau:

Trình bày một bài nghiên cứu, đánh giá của em về các vấn đề sau:

**Câu 1 :** Tìm hiểu, so sánh các phương pháp Optimizer trong huấn luyện mô hình học máy

**Câu 2 :** Tìm hiểu về Continual Learning và Test Production khi xây dựng một giải pháp học máy để giải quyết một bài toán nào đó.

## MỤC LỤC

|  |    |
|--|----|
| LỜI CẢM ƠN .....   | 1  |
| PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN .....   | 3  |
| TÓM TẮT .....  | 4  |
| MỤC LỤC.....   | 5  |
| CHƯƠNG I: TÌM HIỂU, SO SÁNH CÁC PHƯƠNG PHÁP OPTIMIZER TRONG HUẤN LUYỆN MÔ HÌNH HỌC MÁY ..... | 7  |
| 1. Thuật toán tối ưu optimizers.....   | 7  |
| 2. Gradient Descent Optimizer. ....  | 8  |
| 3. Stochastic Gradient Descent Optimizer.....  | 9  |
| 4. Gradient Descent with Momentum. ....  | 10 |
| 5. AdaGrad ( Adaptive Gradient Descent) Optimizer .....                                      | 11 |
| 6. Root Mean Squared Propagation (RMSprop) .....   | 12 |
| CHƯƠNG II: CONTINUAL LEARNING VÀ TEST PRODUCTION KHI XÂY DỰNG MỘT GIẢI PHÁP HỌC MÁY .....    | 15 |
| 1. Continual learning .....  | 15 |
| a. Giới thiệu.....   | 15 |
| b. Một số chiến lược để học trong Continual Learning. ....                                   | 16 |
| c. Đánh giá .....  | 18 |
| 2. Test Production.....  | 18 |
| a. Model Evaluation .....  | 19 |

|                         |                         |    |
|-------------------------|-------------------------|----|
| b.                      | Pre training test.....  | 19 |
| c.                      | Post training test..... | 19 |
| d.                      | API test.....           | 21 |
| e.                      | Đánh giá .....          | 21 |
| TÀI LIỆU THAM KHẢO..... |                         | 22 |



# CHƯƠNG I: TÌM HIỂU, SO SÁNH CÁC PHƯƠNG PHÁP OPTIMIZER TRONG HUẤN LUYỆN MÔ HÌNH HỌC MÁY

## 1. Thuật toán tối ưu optimizers

Trong một mô hình học máy , optimizer là một trong những bước quan trọng nhất.

Một phương pháp Optimizer tốt có thể giúp ta tiết kiệm thời gian , công sức , tiền bạc trong việc huấn luyện mô hình , nhất là đối với những mô hình mà tập huấn luyện rất lớn , khi mà thời gian để train 1 epochs cũng là một con số đáng kể.

Có rất nhiều phương pháp Optimizer khác nhau , trong đó , một số phương pháp phổ biến có thể kể đến như :

- Gradient Descent Optimizer.
- Stochastic Gradient Descent Optimizer.
- Gradient Descent with Momentum.
- AdaGrad ( Adaptive Gradient Descent) Optimizer.
- Root Mean Squared Propagation (RMSprop)

Mỗi phương pháp sẽ có những ưu nhược điểm khác nhau và những bài toán thích hợp tương ứng với nó , tuy nhiên , mục tiêu chung của các phương pháp này đều là tìm ra một bộ các trọng số (weights) mà kết quả đầu ra của mô hình là tốt nhất.

Sau đây , ta sẽ tìm hiểu lần lượt các thuật toán đã nêu ở trên .

## 2. Gradient Descent Optimizer.

Gradient Descent (GD) là một trong những thuật toán tối ưu phổ biến nhất , nó thường được sử dụng như là một thuật toán nền tảng để các thuật toán khác có thể phát triển lên như Stochastic GD, GD with Momentum, AdaGrad, ....

Cách thức hoạt động :

Sau mỗi lần train, thuật toán sẽ thực hiện tính toán gradient của hàm loss theo từng trọng số, sau đó tiến hành cập nhật từng trọng số theo hướng ngược lại với gradient đó theo một giá trị learning rate được xác định trước.

Công thức :

$$x_{t+1} = x_t - \eta f'(x_t)$$

$x_{t+1}$  : Giá trị trọng số mới.

$x_t$  : Giá trị trọng số hiện tại.

$\eta$  : Learning rate.

$f'(x_t)$  : Giá trị đạo hàm riêng của hàm loss theo biến  $x_t$

Ưu điểm :

- Thuật toán tương đối dễ hiểu đối với tất cả mọi người.
- Dễ triển khai và nâng cấp.

Nhược điểm:

- Dễ bị mắc kẹt trong các cực bộ địa phương (Local Minima).
- Nếu chọn learning rate quá lớn sẽ dẫn đến việc không tìm được cực bộ địa phương , nếu chọn quá nhỏ sẽ dẫn đến việc tốc độ hội tụ chậm.
- Không hiệu quả với các bộ dữ liệu lớn , vì số lượng tính toán cần thực hiện là rất lớn , dẫn đến thời gian huấn luyện của mô hình tăng cao.

### 3. Stochastic Gradient Descent Optimizer.

Stochastic Gradient Descent (SGD) là một thuật toán được phát triển dựa trên thuật toán GD ở trên. Khác với GD, khi ở bước tính giá trị hàm loss, thuật toán sẽ tiến hành với toàn bộ tập dữ liệu, thì ở thuật toán SGD, thuật toán sẽ chọn ra một tập dữ liệu con , sau đó thực hiện việc tối ưu trọng số theo bộ dữ liệu được chọn ra đó.

Ưu điểm :

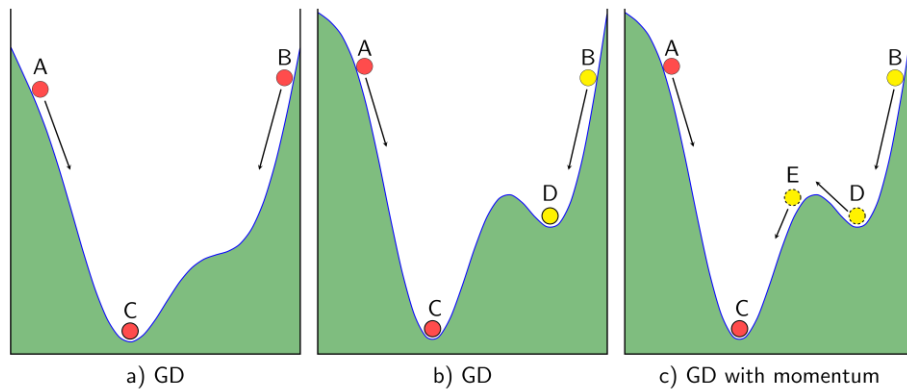
- Tốc độ thực hiện tối ưu hóa nhanh hơn.
- Dễ hiểu và dễ dàng triển khai.

Nhược điểm:

- Dễ bị mắc kẹt trong các cực bộ địa phương (Local Minima).
- Vì chỉ thực hiện tính toán trên một tập dữ liệu con của bộ dữ liệu gốc , vì thế , SGD có thể sẽ không thể đạt được giá trị tối ưu tốt như thuật toán GD.

#### 4. Gradient Descent with Momentum.

Gradient Descent with Momentum là một thuật toán biến thể từ GD , nó áp dụng một nguyên tắc vật lý , đó là mô men quán tính , khi vị trí ban đầu đủ lớn , lúc đi qua một cực bộ địa phương , nếu vận tốc đủ lớn , ta có thể vượt qua cực bộ địa phương đó để tiếp tục đi xuống.



*So sánh GD và GD with Momentum*

Ở đây ta thấy , ở hình b) , với thuật toán GD, khi xuất phát từ B, ta sẽ bị mắc kẹt tại local minima D , tuy nhiên , khi áp dụng momentum , nếu độ cao và dốc của B đủ lớn , thì khi tới D , ta sẽ có một lực quán tính , và nếu quán tính này đủ lớn , ta có thể vượt qua được điểm D để tới điểm E, lúc này ta có thể tới được điểm C là global minima.

Công thức :

$$\theta = \theta - v_t$$

Trong đó :

$\theta$  : giá trị trọng số đang xét.

$v_t$  : giá trị momentum (vận tốc) tại lần train thứ  $t$ . Giá trị này được tính như sau :

$$v_t = \gamma v_{t-1} + \eta \nabla_{\theta} J(\theta)$$

Trong đó :

$v_t$  : giá trị momentum tại lần train  $t$ .

$\gamma$  : hệ số momentum , thường được đặt bằng 0.9

$\eta$  : learning rate.

$\nabla_{\theta} J(\theta)$  : Giá trị đạo hàm riêng của hàm loss theo biến  $\theta$ .

Ưu điểm :

- Vượt qua được một số local minima mà GD mắc phải.
- Có thể đạt được giá trị tốt hơn GD.

Nhược điểm:

- Phải chọn hệ số momentum 1 cách phù hợp , vì có thể xảy ra trường hợp overshooting , khi giá trị tính được vượt qua giá trị cực tiểu và dao động quanh vị trí đó.
- Tính toán phức tạp hơn GD.

## 5. AdaGrad ( Adaptive Gradient Descent) Optimizer

AdaGrad là một biến thể của GD , khác với GD , khi tham số learning rate là cố định , trong thuật toán AdaGrad , learning rate sẽ được thay đổi cho phù hợp với từng trọng số để đạt được một tốc độ hội tụ cao hơn.

Công thức :

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\epsilon I + \text{diag}(G_t)}} \cdot g_t$$

Ưu điểm :

- Learning rate được tự động cập nhật, vì thế ta không cần phải lựa chọn tham số learning thủ công cho từng tham số.
- Cho hiệu quả tốt hơn so với thuật toán GD cổ điển.

Nhược điểm:

- Learning rate sẽ luôn giảm vì  $\text{diag}(G_t)$  sẽ luôn tăng, có thể dẫn đến việc learning rate sẽ rất nhỏ.
- Không hiệu quả bằng một số thuật toán khác như AdaDeta hay Adam.

## 6. Root Mean Squared Propagation (RMSprop)

Thuật toán RMSprop là một thuật toán được cải tiến từ thuật toán AdaGrad , nó giúp giải quyết vấn đề learning rate ( $\frac{\eta}{\sqrt{\epsilon I + \text{diag}(G_t)}}$ ) luôn bị giảm dần trong thuật toán AdaGrad.

Ưu điểm :

- Learning rate được tự động cập nhật, vì thế ta không cần phải lựa chọn tham số learning thủ công cho từng tham số.
- Giải quyết được tình trạng learning rate luôn giảm của thuật toán AdaGrad.

Nhược điểm:

- Có thể dẫn tới hiện tượng overfitting.
- Không hiệu quả bằng một số thuật toán khác như Adam.

| Tên thuật toán                 | Ưu điểm  | Khuyết điểm   | Bài toán thích hợp              |
|--------------------------------|--|---|---------------------------------|
| Gradient Descent               | Đơn giản , dễ triển khai   | Tốc độ hội tụ chậm , nhạy cảm với giá trị learning rate truyền vào.     | Bài toán đơn giản , dữ liệu nhỏ |
| Stochastic Gradient Descent    | Nhanh hơn GD, thích hợp cho bộ dữ liệu lớn                       | Có thể bị kẹt ở các local minima, nhạy cảm với tham số learning rate    | Bài toán với bộ dữ liệu lớn.    |
| Gradient Descent with Momentum | Kết quả hội tụ tốt hơn GD, ít bị mắc vào các local minima hơn GD | Cần phải chọn tham số momentum , có thể dẫn tới hiện tượng overshooting | Bài toán với nhiều local minima |

|         |   |   |   |
|---------|---|---|---|
| AdaGrad | Learning rate được cập nhật tự động   | Có thể dẫn đến việc learning rate quá nhỏ , thời gian hội tụ rất chậm | Bài toán phân loại hình ảnh , nhận dạng giọng nói , xử lí ngôn ngữ tự nhiên     |
| RMSprop | Giải quyết được vấn đề learning rate luôn giảm của AdaGrad, tốc độ hội tụ nhanh hơn AdaGrad | Vẫn nhạy cảm với learning rate, chi phí tính toán có thể rất cao      | Các bài toán tương tự AdaGrad với sự cải tiến về tốc độ hội tụ và learning rate |



## CHƯƠNG II: CONTINUAL LEARNING VÀ TEST PRODUCTION KHI XÂY DỰNG MỘT GIẢI PHÁP HỌC MÁY

### 1. Continual learning

#### a. Giới thiệu.

Được lấy ý tưởng từ việc học của con người , chúng ta có thể học được những điều mới xuyên suốt cuộc đời , và có thể sử dụng lại những kiến thức đó để học những kiến thức mới. Ví dụ , lúc nhỏ ta sẽ được học những thứ đơn giản như con số , các phép toán cộng , trừ , và sau này khi lớn lên , chúng ta có thể áp dụng những kiến thức đã học đó để học những thứ cao cấp hơn như đạo hàm , tích phân,...

Nhờ việc có thể học xuyên suốt như vậy , con người trở nên thông minh hơn theo thời gian.

Continual Learning hay còn được gọi là LifeLong learning , là một phương pháp học được dựa trên cơ sở như vậy. Ở đó , mô hình có thể liên tục học những dữ liệu mới mà không cần phải huấn luyện lại từ đầu, trong khi những dữ liệu cũ vẫn được lưu trữ để có thể dùng được cho sau này.

Trong các mô hình truyền thống , dữ liệu train sẽ được cung cấp từ đầu , bộ dữ liệu này là cố định, nếu muốn nâng cấp mô hình , chúng ta phải thực hiện nâng cấp tập huấn luyện, sau đó thực hiện train mô hình lại từ đầu. Trong thực tế , dữ

liệu của chúng ta luôn luôn thay đổi từng ngày , từng giờ. Vì thế , việc sử dụng các mô hình truyền thống vào thực tế là khá tốn kém.

Ví dụ :

Trong một mô hình phân loại email spam , mô hình này cần phải liên tục học hỏi để có thể nhận diện thêm những loại email mới. Nếu sử dụng những loại mô hình truyền thống , sẽ tốn rất nhiều chi phí , vì lượng dữ liệu rất lớn , khi có những loại email mới cần phân loại, ta phải tốn một lượng tài nguyên đáng kể để train lại từ đầu mô hình này nhằm cập nhật những dữ liệu mới đó, và việc train này phải diễn ra liên tục để đáp ứng được nhu cầu thực tế rằng dữ liệu mới xuất hiện liên tục.

Vì thế , một hướng tập cận mới đã ra đời , đó là Continual Learning, nó giúp ta có thể học được những dữ liệu mới mà không cần phải thực hiện huấn luyện lại mô hình.

#### **b. Một số chiến lược để học trong Continual Learning.**

Catastrophic forgetting hay còn gọi là hiện tượng “Quên nghiêm trọng”, là hiện tượng mà ở đó , máy sẽ quên một số tri thức của nhiệm vụ trước đó, và điều đó sẽ làm giảm hiệu suất của mô hình , vì có thể mô hình sẽ gặp lại một nhiệm vụ cũ , nhưng vì tri thức đó đã bị quên , vì thế phải tiến hành học lại tri thức đó.

Đây là một hiện tượng hết sức phổ biến khi huấn luyện một mô hình học liên tục (Continual Learning) mà nếu không cẩn thận khi huấn luyện, ta rất dễ sẽ mắc phải hiện tượng này. Sau đây là một số chiến lược học liên tục và hạn chế tình trạng Catastrophic forgetting.

i. Dùng một bộ nhớ để lưu lại những thông tin đã học.

Đây là một cách dễ dàng triển khai nhất , một bộ nhớ đủ lớn sẽ giúp mô hình của chúng ta có thể lưu lại tất cả những gì nó đã học.

Tuy nhiên , khi áp dụng phương pháp này , mỗi lần học , sẽ cần thêm một vùng nhớ mới để có thể lưu trữ những thông tin mới nạp vào, cách làm như vậy là cực kì tốn kém và thiếu thực tế. Vì số lượng thông tin nạp vào là rất lớn , và dung lượng bộ nhớ thì có hạn , ta không thể cung cấp một bộ nhớ vô hạn để mô hình có thể nhớ tất cả những gì nó học được.

ii. Học liên tục dựa vào hiệu chỉnh

Là một trong những giải pháp kinh điển trong việc giải quyết vấn đề Castatropic Forgetting.

Một phương pháp tiêu biểu của hướng tiếp cận này là EWC (Elastic Weight Consolidation).

Ý tưởng của phương pháp này là mỗi khi mô hình học một kiến thức mới , mô hình sẽ đánh giá mức độ quan trọng của từng thành phần để đạt được kiến thức mới , nếu thành phần càng quan trọng , độ ưu tiên sẽ càng cao , độ ưu tiên càng cao sẽ giúp cho thành phần đó ít bị biến đổi trong những lần học tiếp theo.

iii. Rehearsal-base Continual Learning

Là một hướng tiếp cận Continual Learning mà trong đó , dữ liệu cũ sẽ liên tục được tái sử dụng. Thay vì chỉ sử dụng những dữ liệu mới,

chúng ta sẽ kết hợp chúng với những dữ liệu cũ để huấn luyện mô hình , nhờ đó, mô hình sẽ duy trì và cập nhật được những dữ liệu cũ này.

Tuy nhiên, cũng giống như giải pháp đầu tiên đã đề cập , hướng tiếp cận này cũng yêu cầu chúng ta phải cung cấp một nguồn tài nguyên để tiến hành lưu trữ dữ liệu , có thể không khả thi trong một số trường hợp.

### **c. Đánh giá**

Qua những điều trên , ta thấy được rằng , Continual Learning là một hướng tiếp cận học máy rất hiệu quả.

Nhờ khả năng học tập xuyên suốt của mình , một model thuộc Continual Learning , việc xây dựng và nâng cấp một mô hình có thể diễn ra một cách nhanh chóng và tiết kiệm chi phí.

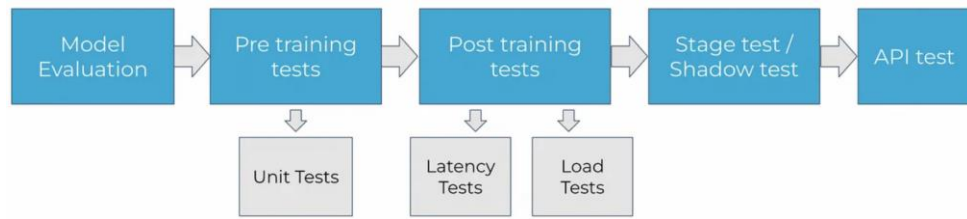
Tuy nhiên , mô hình này vẫn gặp phải nhiều hạn chế , tiêu biểu nhất là vấn đề Catastrophic Forgetting, nó sẽ làm cho mô hình bị giảm đi sự hiệu quả , nếu vấn đề nghiêm trọng , có thể phải tiến hành huấn luyện lại mô hình.

## **2. Test Production**

Test Production là một quá trình kiểm tra tính ổn định và hiệu suất của một giải pháp học máy trước khi triển khai vào môi trường thực tế.

Mục tiêu của test production là đảm bảo rằng mô hình hoạt động đúng đắn , tin cậy và đáp ứng được yêu cầu thực tế.

Các bước để test một mô hình machine learning.



*Quy trình test một mô hình*

#### **a. Model Evaluation**

Đánh giá mô hình là bước đầu tiên của quy trình kiểm thử mô hình, mỗi loại mô hình sẽ có những metrics tương ứng :

1. Mô hình dạng classification : Accuracy, Precision, F1 Score ,...
2. Mô hình dạng Regression : RMSE, MSE, MAE,...

#### **b. Pre training test**

Trước thực hiện train mô hình , chúng ta cần thực hiện một số unit test để đảm bảo rằng dữ liệu đưa vào mô hình đúng dạng , đủ số lượng , không có dữ liệu nhiễu,...

#### **c. Post training test**

##### **1. Batch Training và Online Training**

Batch Training và Online Training là hai dạng mô hình học máy , trong đó:

- Batch Training :

Là những dạng mô hình truyền thống , tức là chúng ta phải thực hiện huấn luyện mô hình một cách hoàn chỉnh , sau đó sẽ thực hiện dự đoán dựa trên mô hình đã học.

- Online Learning :

Là quá trình học mà mô hình có thể liên tục học những dữ liệu mới mà không cần phải huấn luyện lại toàn bộ mô hình. Nhờ đó giúp tiết kiệm thời gian , chi phí , tiền bạc.

Mô hình này rất cần thiết với những hệ thống dự đoán thời gian thực như phân loại email,...

Continual Learning là một loại mô hình thuộc dạng Online Learning này.

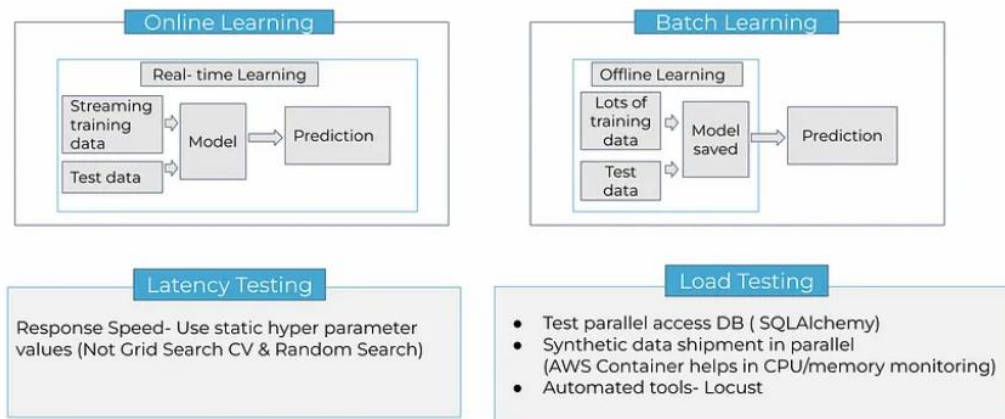
## 2. Post training test với mô hình dạng batch training

Đối với mô hình batch training , ta sẽ thực hiện kiểm tải mô hình.

Kiểm tải là quá trình ta kiểm tra xem hệ thống có thể xử lí cùng lúc bao nhiêu yêu cầu

## 3. Post training test với mô hình dạng online training

Đối với mô hình dạng online training , ta cần kiểm tra tốc độ phản hồi của mô hình, nếu mô hình phản hồi quá chậm, chúng ta cần có các giải pháp để tăng tốc mô hình hoặc thậm chí phải thay đổi mô hình.



*Post training test với online learning và batch learning*

#### d. API test

Là bước kiểm thử cuối cùng , ở đây , chúng ta sẽ kiểm tra thực sự người dùng cuối sẽ nhận được câu trả lời như thế nào từ mô hình của chúng ta.

#### e. Đánh giá

Test Production là một bước rất quan trọng khi phát triển một mô hình học máy ra thị trường , nó sẽ giúp ta sớm khắc phục được những lỗi có thể gặp phải , cũng như đảm bảo mô hình của chúng ta sẽ chạy một cách mượt mà nhất cũng những đưa ra những kết quả tốt nhất.

## **TÀI LIỆU THAM KHẢO**

### **1. Tiếng Việt**

Hiểu sâu về các thuật toán tối ưu – Trần Trung Trực (Viblo)

Robot có thể học suốt đời không? – Thân Quang Khoát

### **2. Tiếng Anh**

Introduction to Continual Learning – Continual AI

Introduction to Continual Learning – Giang Nguyen

Machine Learning Model Testing for Production - Shivika K Bisen