

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



BÁO CÁO CUỐI KÌ

MÔN HỌC: NHẬP MÔN HỌC MÁY

Người hướng dẫn: **PGS.TS. LÊ ANH CƯỜNG**

Người thực hiện: **NGUYỄN THÁI HÒA – 52100413**

Lớp : 21050401

Khoá : 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**



BÁO CÁO CUỐI KÌ

MÔN HỌC: NHẬP MÔN HỌC MÁY

Mã môn học: 503044

Người hướng dẫn: **PGS.TS. LÊ ANH CƯỜNG**

Người thực hiện: **NGUYỄN THÁI HÒA – 52100413**

Lớp : 21050401

Khoá : 25

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023

LỜI CẢM ƠN

Môn học Nhập môn Học máy thực sự rất bổ ích, đã bổ trợ thêm cho em rất nhiều kiến thức liên quan đến chuyên ngành của mình. Em xin cảm ơn sự quan tâm của nhà trường và khoa CNTT khi đã đưa bộ môn vào giảng dạy. Đặc biệt em xin gửi lời cảm ơn chân thành đến giảng viên bộ môn – thầy PGS.TS. Lê Anh Cường đã luôn ân cần giảng dạy, giải đáp tận tình những thắc mắc của chúng em, giúp chúng em có thể hiểu và làm bài một cách đơn giản nhất.

Bài báo cáo này là minh chứng rõ nhất cho những cố gắng, nỗ lực học tập của em. Tuy vậy sự tiếp thu kiến thức của em chỉ mới ở mức cơ bản và còn nhiều thiếu sót, kính mong thầy có thể cho em thêm nhận xét và góp ý để bài báo cáo có thể hoàn thiện hơn. Em xin chân thành cảm ơn!

BÁO CÁO ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Tôi xin cam đoan đây là sản phẩm đồ án của riêng tôi và được sự hướng dẫn của thầy PGS.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 đồ 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 đồ 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 18 tháng 12 năm 2023

Tác giả

(ký tên và ghi rõ họ tên)

Hòa

Nguyễn Thái Hòa

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

Phần xác nhận của GV hướng dẫn

Tp. Hồ Chí Minh, ngày tháng năm
(kí và ghi họ tên)

Phần đánh giá của GV chấm bài

Tp. Hồ Chí Minh, ngày tháng năm
(kí và ghi họ tên)

TÓM TẮT

Đối với nội dung tìm hiểu cá nhân trong bài một, em tiến hành tìm hiểu, nghiên cứu và đưa ra đánh giá, so sánh các phương pháp Optimizer trong huấn luyện mô hình học máy. Ngoài ra, cũng tiến hành 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
TÓM TẮT	4
MỤC LỤC.....	5
DANH MỤC CÁC CHỮ VIẾT TẮT	6
DANH SÁCH CÁC BẢNG BIỂU, HÌNH VẼ	7
PHẦN 1: CÁC PHƯƠNG PHÁP OPTIMIZER	8
I. Tổng quan:.....	8
II. Mục tiêu:	8
III. Tìm hiểu và so sánh các phương pháp Optimizer:	8
3.1 Tổng quan về Optimizer:	8
3.2 Các phương pháp Optimizer phổ biến.....	9
3.3 So sánh và đánh giá:	16
IV. Continual Learning và Test Production trong Giải Quyết Bài Toán	
Xác Định.....	17
4.1 Continual Learning:	17
4.2 Test Production:	22

DANH MỤC CÁC CHỮ VIẾT TẮT

DANH SÁCH CÁC BẢNG BIỂU, HÌNH VẼ

Hình 3. 1 Hình ảnh minh họa các trường hợp khi thay đổi learning rate	10
Hình 3. 2 Hình ảnh minh họa so sánh giữa SGD và GD	11
Hình 3. 3 Hình ảnh so sánh độ hiệu quả giữa các phương pháp	16
 Hình 4. 1 Hình ảnh qui trình Continual learning	20
Hình 4. 2 Hình ảnh quá trình huấn luyện của mô hình	24
Hình 4. 3 Các phương pháp tối ưu	24
Hình 4. 4 Kết quả so sánh	25
Hình 4. 5 Mô phỏng ContinualLearning và test production	25
Hình 4. 6 Huấn luyện mô hình lần thứ nhất	26
Hình 4. 7 Huấn luyện mô hình lần thứ hai	26

PHẦN 1: CÁC PHƯƠNG PHÁP OPTIMIZER

I. Tổng quan:

Trong học máy hiện nay các phương pháp Optimizer và Continual Learning luôn là những vai trò quan trọng trong việc huấn luyện 1 mô hình, đặc biệt là đối với ngày nay dữ liệu ngày càng phức tạp đòi hỏi mô hình càng được tối ưu về hiệu suất hơn. Khi áp dụng Optimizer và Continual Learning sẽ đem tới 1 số lợi ích như là tăng hiệu suất của mô hình để có khả năng mở rộng, tăng khả năng thích ứng với các loại dữ liệu khác nhau, tổng quát hóa dữ liệu giúp mô hình luôn đảm bảo khả năng tổng quát dữ liệu mới mà không đánh mất dữ liệu cũ,... Sự kết hợp giữa các phương pháp là chìa khóa để xây dựng ra một mô hình học máy tốt mà còn có thể được ứng dụng rộng rãi.

II. Mục tiêu:

Tìm hiểu, nghiên cứu, so sánh và đánh giá các phương pháp Optimizer trong huấn luyện mô hình học máy, cũng như là Continual Learning và Test production trong xây dựng mô hình học máy.

III. Tìm hiểu và so sánh các phương pháp Optimizer:

3.1 Tổng quan về Optimizer:

Huấn luyện mô hình học máy là quá trình tìm ra các tham số đầu vào để mô hình có thể tổng quát thông tin và đưa ra dự đoán. Vậy việc chọn tham số đầu vào như thế nào là hợp lý là vấn đề của Optimizer, khi này optimizer đóng vai trò rất quan trọng trong

việc cải thiện hiệu suất của mô hình và giúp mô hình biểu diễn được ở nhiều dạng dữ liệu khác nhau.

3.2 Các phương pháp Optimizer phổ biến

- Gradient Descent (GD):

Gradient Descent là một thuật toán tối ưu thường được sử dụng để đào tạo các mô hình học máy. Dữ liệu được đào tạo sẽ giúp mô hình học qua nhiều lần và tính toán độ chính xác thông qua mỗi lần cập nhật tham số cho đến khi cost function tiến gần về 0.

a) Nguyên lý hoạt động:

- Tối ưu: GD tối ưu hàm mất mát thường được kí hiệu là $J(\theta)$, trong đó θ là tập hợp các tham số.
- Đạo hàm: tiếp tục đạo hàm của hàm mất mát tại điểm hiện tại để xác định hướng và độ lớn cần điều chỉnh các tham số.
- Learning rate: Là 1 tham số quan trọng của GD dùng để xác định kích thước cập nhật tham số.

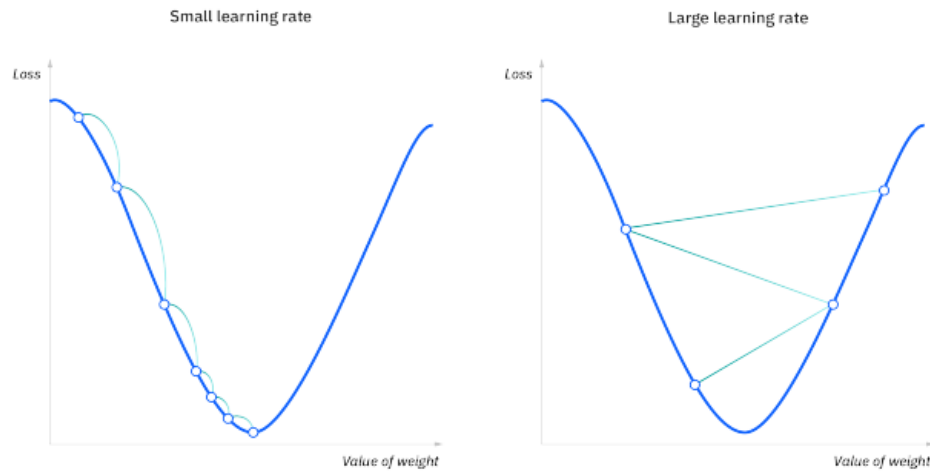
b) Công thức:

$$\theta := \theta - \alpha \frac{\partial}{\partial \theta} J(\theta)$$

Trong đó, GD có các tham số:

- α là learning rate
- $\frac{\partial}{\partial \theta} J(\theta)$ là đạo hàm của hàm $J(\theta)$, có tác dụng tính độ dốc của điểm hiện tại.

Qua công thức trên GD sẽ lặp đi lặp lại đến chừng nào tìm được điểm trung thấp nhất của hàm $J(\theta)$ vì qua mỗi vòng lặp $J(\theta)$ sẽ càng chính xác.



Hình 3. 1 Hình ảnh minh họa các trường hợp khi thay đổi learning rate

c) Ưu điểm:

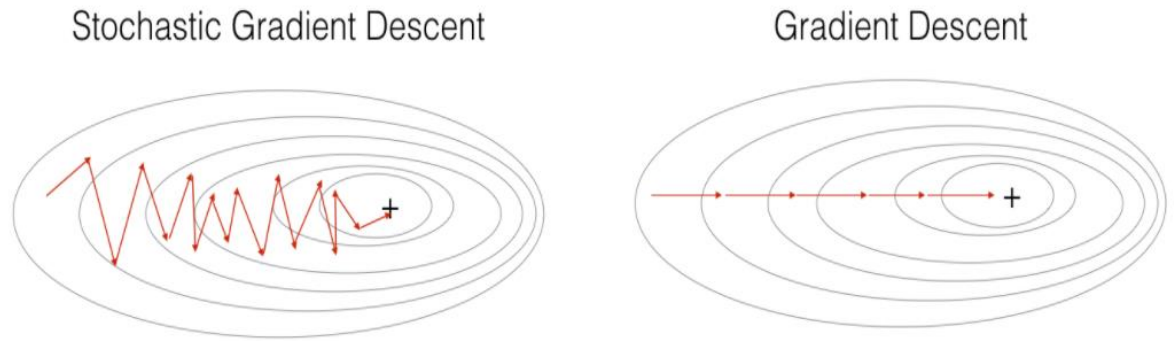
Thuật toán đơn giản dễ sử dụng, giải quyết được vấn đề tối ưu sau nhiều lần cập nhật lại trọng số.

d) Nhược điểm:

- Phụ thuộc vào các tham số đầu vào
- Learning rate lớn sẽ làm cho thuật toán khó hiệu quả.

- Stochastic Gradient Descent (SGD):

Stochastic Gradient Descent là một dạng của Gradient Descent. Với gradient descent thì mỗi khi cập nhật thường thì sẽ cập nhật lại tham số 1 lần nhưng đối với stochastic gradient descent thì trong mỗi lần cập nhật là sẽ có N điểm dữ liệu và sẽ cập nhật trọng số N lần. Như ta đã thấy nếu như cập nhật 1 lần nhiều trọng số thì sẽ làm giảm hiệu suất cập nhật nhưng bù lại sẽ chỉ cần ít lần cập nhật hơn so với dạng gradient descent cơ bản.



Hình 3. 2 Hình ảnh minh họa so sánh giữa SGD và GD

a. Nguyên lý hoạt động:

Thay vì cập nhật dựa trên toàn bộ tập dữ liệu như Gradient Descent thì SGD chỉ sử dụng 1 lượng nhỏ ngẫu nhiên của bộ dữ liệu để tính toán và cập nhật trọng số.

Vì vậy bài toán SGD sinh ra để giải quyết bài toán cập nhật lại trọng số thay vì cập nhật toàn bộ khi có dữ liệu được thêm vào thì chỉ cần cập nhật lại 1 điểm dữ liệu đó thôi khiến tăng hiệu suất.

b. Ưu điểm: giải quyết được các tập dữ liệu lớn hơn GD.

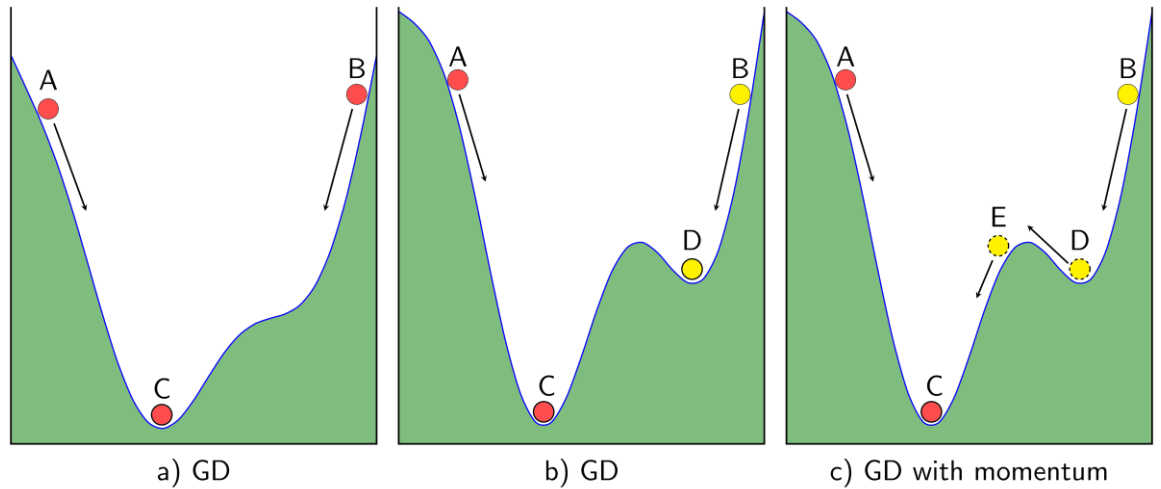
c. Nhược điểm: vẫn còn phụ thuộc vào các tham số ban đầu.

- **Momentum**

Momentum sinh ra để giải quyết vấn đề optimizer trong học máy khi mà chúng ta tối ưu hóa là sẽ tìm điểm cực tiểu global nhưng chúng ta sẽ gặp vấn đề là bị mắc kẹt ở điểm cực tiểu local để tránh mắc phải vấn đề này chúng ta sẽ sử dụng momentum.

a. Nguyên lý hoạt động:

Momentum hoạt động bằng cách tích lũy 1 vector vận tốc để giữ cho quá trình cập nhật trọng số được hướng về 1 hướng để có thể giải quyết vấn đề bị mắc kẹt ở cực tiểu local.



b. Công thức:

$$v_{t+1} = \alpha \cdot v_t + \eta \cdot J(\theta_t)$$

$$\theta_{t+1} = \theta_t - v_{t+1}$$

Trong đó:

- v_t là vận tốc tại t
- θ_t là vector trọng số tại t
- η là learning rate
- $J(\theta_t)$ là hàm mất mát.

c. Ưu điểm: giải quyết được vấn đề điểm cực tiểu local.

d. Nhược điểm: Mặc dù hội tụ được tại điểm cực tiểu global nhưng vẫn mất nhiều thời gian để xử lý khi gần hội tụ.

• Adagrad:

Adagrad là phương pháp tối ưu trong học máy được thiết kế để có thể tự điều chỉnh tham số learning rate phụ thuộc vào các gradient trước đó. Phương pháp này sẽ cho giảm learning rate khi tham số có gradient lớn và tăng learning rate cho các tham số có gradient nhỏ.

a. Công thức:

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,(i,i)} + \epsilon}} \cdot g_{t,i}$$

Trong đó:

- $\theta_{t,i}$ là giá trị của tham số thứ i tại t
 - η là learning rate
 - $g_{t,i}$ là gradient của tham số thứ i tại t
 - $G_{t,(i,i)}$ là tổng bình phương của các gradient trước đó cho tham số thứ i
 - ϵ là tham số chống chia cho 0
- b. Ưu điểm: Tránh được việc điều chỉnh learning rate bằng tay, mặc định cho 1 giá trị thuật toán sẽ tự điều chỉnh.
- c. Nhược điểm: Nếu tổng bình phương biến thiên lớn dần theo thời gian learning rate sẽ giảm dần khiến, việc tối ưu trở nên chậm đi.

• RMSProp:

RMSprop là một phương pháp tối ưu trong học máy, được dùng để giải quyết vấn đề tổng bình phương của Adagard bằng cách tính toán tổng bình phương theo thời gian. RMSprop giúp tối ưu và thích ứng tốt hơn với độ lệch và tỷ lệ khác nhau giữa các tham số.

a. Công thức:

- Công thức cập nhật tổng bình phương:

$$G_{t,(i,i)} = \gamma \cdot G_{t-1,(i,i)} + (1 - \gamma) \cdot g_{t,i}^2$$

Trong đó:

- $G_{t,(i,i)}$ là tổng bình phương của các gradient trước đó cho tham số thứ i
- $g_{t,i}$ là gradient của tham số thứ i tại t

- γ là hệ số giảm ảnh hưởng (decay rate)

- Công thức cập nhật trọng số:

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,(i,i)} + \epsilon}} \cdot g_{t,i}$$

Trong đó:

- $\theta_{t,i}$ là giá trị của tham số thứ i tại t
- η là learning rate
- $g_{t,i}$ là gradient của tham số thứ i tại t
- $G_{t,(i,i)}$ là tổng bình phương của các gradient trước đó cho tham số thứ i
- ϵ là tham số chống chia cho 0

b. Ưu điểm: Giải quyết learning rate giảm dần của phương pháp Adagard

c. Nhược điểm: Vấn đề mắc kẹt ở các cực tiểu local bởi vì không có vector vận tốc.

- **Adam:**

Adam (adaptive Moment Estimation) là một phương pháp tối ưu trong học máy với sự kết hợp của RMSprop và Momentum. Adam được sử dụng để cân bằng giữa việc giảm độ giao động của các trọng số và sử dụng các thông tin về gradient từ các bước cập nhật trước đó.

a. Công thức:

- Momentum:

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$$

Trong đó:

- m_t là vector vận tốc tại t
- β_1 là hệ số giảm của Momentum

- g_t là gradient của hàm mất mát tại t

▪ RMSProp:

$$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot (g_t^2)$$

Trong đó:

- v_t là vector RMSprop tại t
- β_2 là hệ số giảm cho RMSprop
- g_t là gradient của hàm mất mát tại t

Bias Correction: bước này là bước kiểm soát bias đối với và để đảm bảo tính chính xác của ước lượng đối với những bước đầu tiên.

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}$$

Cập nhật lại trọng số:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} \cdot \hat{m}_t$$

Trong đó:

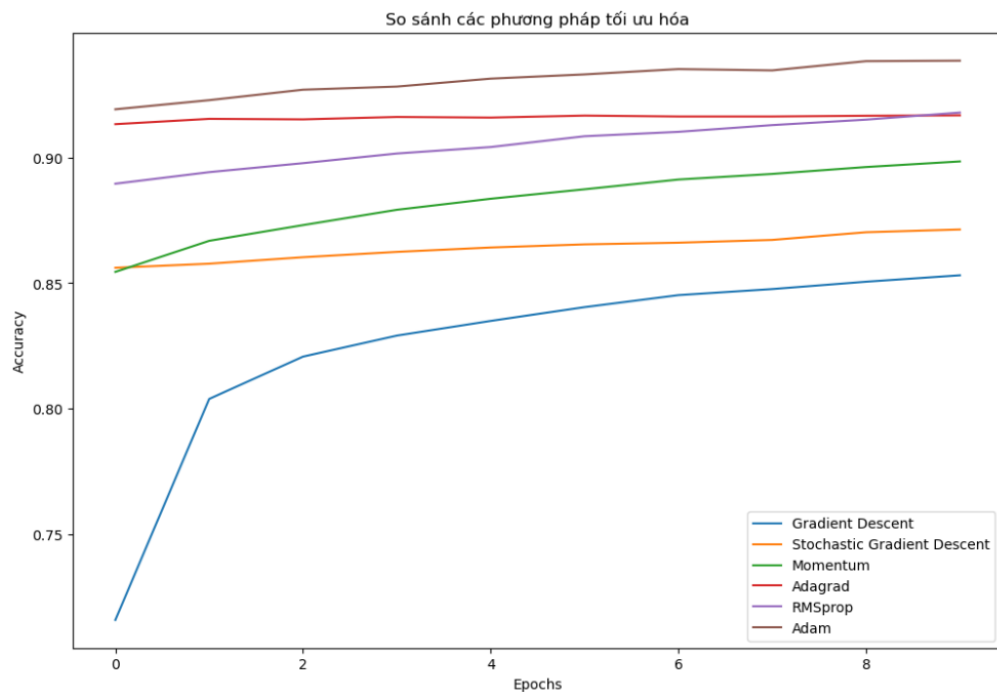
- θ_t là vector trọng số tại t
- η là learning rate
- ϵ là tham số chống chia cho 0
- v_t là vector RMSprop tại t
- m_t là vector vận tốc tại t

b. Ưu điểm:

- Hiệu quả và thường được sử dụng rộng rãi trong học máy.

- Tự động điều chỉnh learning rate giúp hiệu quả hơn những phương pháp cố định learning rate.
- c. Nhược điểm:
- Nhạy cảm với nhiễu trong dữ liệu.
 - Vấn đề overfitting khi quá hiệu quả.

3.3 So sánh và đánh giá:



Hình 3.3 Hình ảnh so sánh độ hiệu quả giữa các phương pháp

Trong đó, chúng ta sử dụng thư viện keras để tạo model với 1 lớp.

```
# Model
model = tf.keras.models.Sequential([
    tf.keras.layers.Dense(128, activation='relu', input_shape=(784,)),
    tf.keras.layers.Dense(num_classes, activation='softmax')
])
```

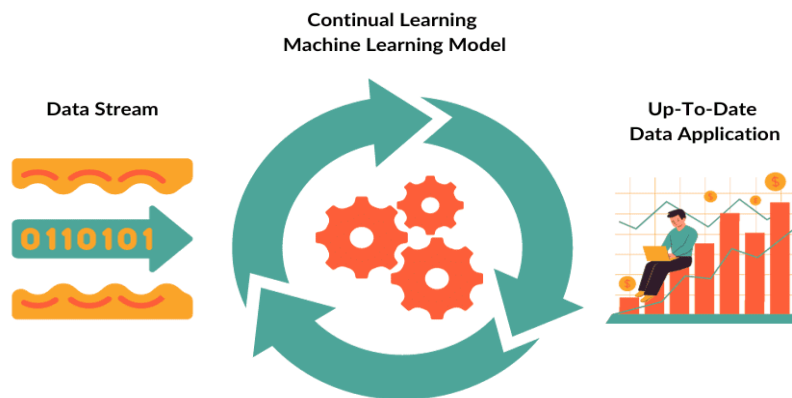
Sau đó huấn luyện mô hình với epochs = 10

```
# Hàm huấn luyện
def train(optimizer):
    model.compile(optimizer=optimizer, loss='categorical_crossentropy', metrics=['accuracy'])
    history = model.fit(x_train, y_train, epochs=10, batch_size=64, validation_split=0.2, verbose=0)
    return history.history['accuracy']
```

IV. Continual Learning và Test Production trong Giải Quyết Bài Toán Xác Định

4.1 Continual Learning:

Continual learning trong học máy còn được biết đến là 1 quá trình trong đó mô hình học từ các dữ liệu mới mà không cần đào tạo lại. Nếu ở với các phương pháp truyền thống thì mô hình được đào tạo trên 1 tập dữ liệu tĩnh và được đào tạo lại định kỳ, cập nhật lại các tham số liên tục để phản ánh sự thay đổi trong dữ liệu. Nhưng trong continual learning mô hình sẽ tự đào tạo lại bằng cách học tập từ lần lặp mới nhất và cập nhật sự thay đổi khi có dữ liệu mới.



a. Các loại Continual Learning:

Có nhiều loại Continual Learning để áp dụng cho mô hình. Các loại phổ biến bao gồm incremental learning, transfer learning và lifelong learning.

- Incremental learning: là một hình thức của continual learning trong đó mô hình được huấn luyện dần dần trên các dữ liệu mới mà không quên đi dữ liệu cũ. Các đặc điểm của phương pháp này:

- Khả năng mở rộng: Mô hình có thể mở rộng để train các dữ liệu mới mà không cần học lại từ đầu.
- Quản lý dữ liệu cũ: Đảm bảo việc không mất đi thông tin từ dữ liệu đã học bằng cách thông qua việc kết hợp dữ liệu mới và cũ.
- Tối ưu hóa: Cập nhật mô hình nhanh chóng khi có sự thay đổi giúp tối ưu hóa hiệu suất
- Transfer learning: Là một hình thức của continual learning mà mô hình được huấn luyện trước trên 1 tập dữ liệu lớn, sau đó dữ liệu đó được dùng để giải quyết 1 bài toán cụ thể hoặc 1 tập dữ liệu mới. Các đặc điểm của phương pháp này:
 - Pre-training và Fine-tuning: Mô hình sẽ được huấn luyện trên 1 tập dữ liệu lớn trước khi giải quyết 1 bài toán cụ thể.
 - Chuyển giao kiến thức: Kiến thức học từ một tập dữ liệu lớn có thể được sử dụng để giúp quá trình học trên dữ liệu mới.
 - Hiệu suất cao: Khi được học trên dữ liệu mới có ít dữ liệu mô hình sẽ có hiệu quả cao hơn.
- Lifelong Learning: là một hình thức của Continual learning mà mô hình được huấn luyện trên nhiều tập dữ liệu khác nhau theo thời gian và phải giải quyết nhiều nhiệm vụ khác nhau. Các đặc điểm chính bao gồm:
 - Học liên tục: Mô hình không chỉ học từ các dữ liệu mới mà còn giải quyết nhiều nhiệm vụ theo thời gian.
 - Quản lý đa nhiệm: Có khả năng quản lý và kết từ các nhiệm vụ khác nhau.
 - Chuyển giao thông tin: Thông tin từ mỗi nhiệm vụ có thể được sử dụng để hỗ trợ cho các nhiệm vụ mới.

Mỗi loại Continual Learning có các ưu và nhược điểm riêng. Incremental Learning thích hợp cho các mô hình thêm dữ liệu liên tục mà không cần huấn luyện lại toàn bộ. Transfer Learning phù hợp khi có sẵn một mô hình đã được đào tạo và áp dụng

qua một dữ liệu cụ thể. Lifelong learning là lựa chọn khi mô hình cần đào tạo liên tục và làm nhiều nhiệm vụ một lúc trong thời gian dài.

b. Quy trình của Continual Learning

Quy trình của continual learning là sự phát triển của mô hình học máy cơ bản.

Do đó sẽ có các bước cơ bản:

- Tiền xử lý dữ liệu: Làm sạch và chuẩn hóa dữ liệu.
- Lựa chọn mô hình: Lựa chọn mô hình phù hợp với bài toán.
- Điều chỉnh tham số: Tùy chỉnh các tham số ảnh hưởng đến hiệu suất của mô hình.
- Huấn luyện mô hình: Tiến hành quá trình học từ dữ liệu và huấn luyện mô hình.
- Triển khai và giám sát: Triển khai và thực hiện sự quan sát mô hình.

Trong Continual learning cần phải bổ sung 2 bước cần thiết để đảm bảo quá trình được học từ các luồng dữ liệu mới một cách hiệu quả:

- Luyện tập dữ liệu (Data Rehearsal): Việc thực hiện lại mô hình với dữ liệu mới, giúp mô hình cập nhật thông tin từ dữ liệu mới.
- Thiết lập chiến lược Continual learning (Continual Learning Strategy): Xây dựng kế hoạch và chiến lược đảm bảo mô hình được học liên tục từ dữ liệu mới 1 cách hiệu quả.

The Continuous Learning Process



Hình 4. 1 Hình ảnh qui trình Continual learning

c. Ưu điểm của continual learning:

Continual learning có thể hữu ích với nhiều loại mô hình đặc biệt là trong các trường hợp liên quan tới dữ liệu thay đổi nhanh chóng. Sau đây là các ưu điểm so với phương pháp truyền thống:

- Tổng quát hóa: Mô hình dựa vào cả dữ liệu cũ và dữ liệu mới giúp tăng khả năng tổng quát hóa.
- Giữ lại thông tin: Mô hình sử dụng dữ liệu đã học trước đó giúp tích lũy thông tin theo thời gian. Việc này giúp mô hình không quên thông tin đã học và giúp huấn luyện tốt hơn với dữ liệu mới.
- Khả năng thích ứng: Mô hình sử dụng khả năng continual learning có khả năng thích ứng với dữ liệu mới, điều này giúp tăng khả năng dự đoán và linh hoạt.

Continual learning đem đến sự linh hoạt và khả năng thích ứng với mô hình học máy, đặc biệt với những bộ dữ liệu thay đổi liên tục.

d. Nhược điểm của continual learning

Phương pháp continual learning mặc dù có rất nhiều ưu điểm nhưng cũng đi kèm 1 số nhược điểm như sau:

- Chi phí: Phương pháp continual learning phức tạp hơn về mặt tính toán với những phương pháp truyền thống vì mô hình cần liên tục điều chỉnh dựa trên dữ liệu mới.
- Quản lý mô hình: Mỗi khi dữ liệu được cập nhật thì mô hình mới được tạo ra. Vậy nên cần quản lý các mô hình 1 cách hiệu quả để tránh ảnh hưởng đến hiệu suất.
- Biến động trong dữ liệu: Đối với continual learning phải xử lý 1 lượng lớn dữ liệu, nên phương pháp này có rủi ro mất khả năng dự đoán khi các đặc trưng của dữ liệu thay đổi đột ngột.

e. Ứng dụng của Continual learning

Continual learning thích hợp cho các ứng dụng liên quan đến dữ liệu được cập nhật liên tục. Các ứng dụng phổ biến hiện nay:

- Thị giác máy tính: Tính động trong hình ảnh làm cho continual learning trở nên phổ biến trong việc huấn luyện nhận diện và phân loại thông tin hình ảnh.
- Bảo mật mạng: Continual learning được triển khai để đảm bảo giám sát liên tục trong cơ sở hạ tầng, nhằm mục đích phát hiện lừa đảo, xâm nhập hệ thống, ...
- Chăm sóc sức khỏe: Áp dụng continual learning trong vấn đề sức khỏe nhằm cải thiện khả năng chuẩn đoán bệnh tật.



4.2 Test Production:

a. Các loại Test

Trong quá trình phát triển mô hình, có 5 loại test chính được sử dụng trong các giai đoạn khác nhau:

- Unit test:
 - Mục tiêu: Unit test trên các thành phần nhỏ và mỗi thành phần có 1 nhiệm vụ nhất định.
 - Ví dụ: Kiểm thử trên các hàm.
- Intergration Test:
 - Mục tiêu: Kiểm thử sự kết hợp giữa các thành phần.
 - Ví dụ: Kiểm thử giữa các mô hình với nhau.
- System Test:
 - Mục tiêu: Kiểm thử hệ thống đảm bảo đầu ra đúng với đầu vào.
 - Ví dụ: Áp dụng trên toàn bộ hệ thống để kiểm tra tất cả chức năng.
- Acceptance test:
 - Mục tiêu: Xác nhận rằng các yêu cầu của khách hàng đã được đáp ứng.

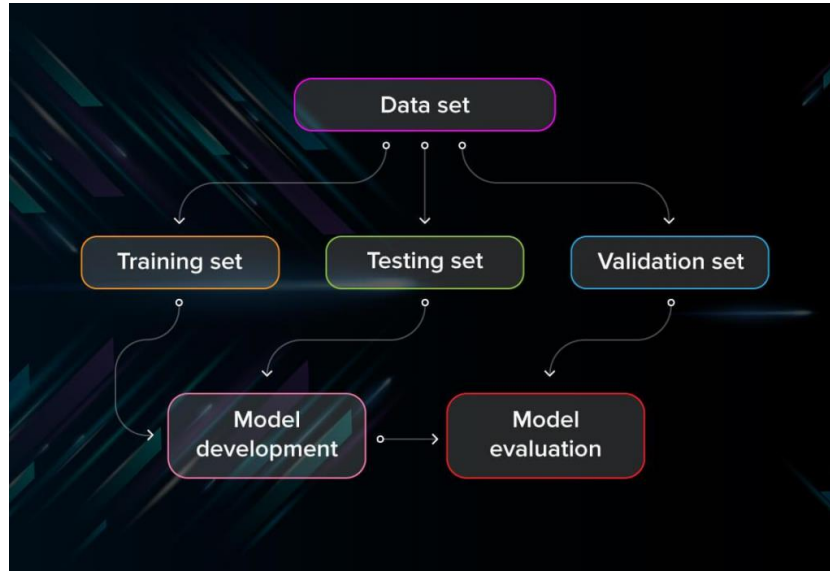
- Ví dụ: Đảm bảo rằng hệ thống đáp ứng đúng yêu cầu mong đợi của người dùng.
- Regression Test:
 - Mục tiêu: Kiểm thử dựa trên các lỗi trước đó đảm bảo không xảy ra nữa.
 - Ví dụ: Đảm bảo sau khi đã sửa chữa thì sẽ không gây ra lỗi trước đó.

b. So sánh Evaluation và Testing Production

Evaluation tập trung vào việc đánh giá hiệu suất của mô hình bao gồm các chỉ số và đồ thị hiệu suất. Tuy nhiên, Evaluation không cung cấp thông tin về các nguyên nhân sâu rộng về sự sai sót trong quá trình huấn luyện như testing production.

Các tính chất của Testing Production:

- Phát hiện sự thay đổi: Trong quá trình huấn luyện mô hình phải đối mặt với nhiều sự thay đổi và Test production giúp phát hiện sự thay đổi đột ngột của mô hình.
- Kiểm soát hiệu suất: Testing giúp duy trì sự ổn định của mô hình đặc biệt là khi có những sự thay đổi trong dữ liệu.
- Thông báo nguyên nhân của lỗi: Khi xảy ra lỗi Testing production không chỉ đo lường chỉ số mà còn giúp chúng ta phân tích nguyên nhân xảy ra lỗi.
- Hỗ trợ continual learning: testing đóng vai trò quan trọng trong Continual learning hỗ trợ quá trình thích nghi với dữ liệu mới mà không làm giảm hiệu suất mô hình.



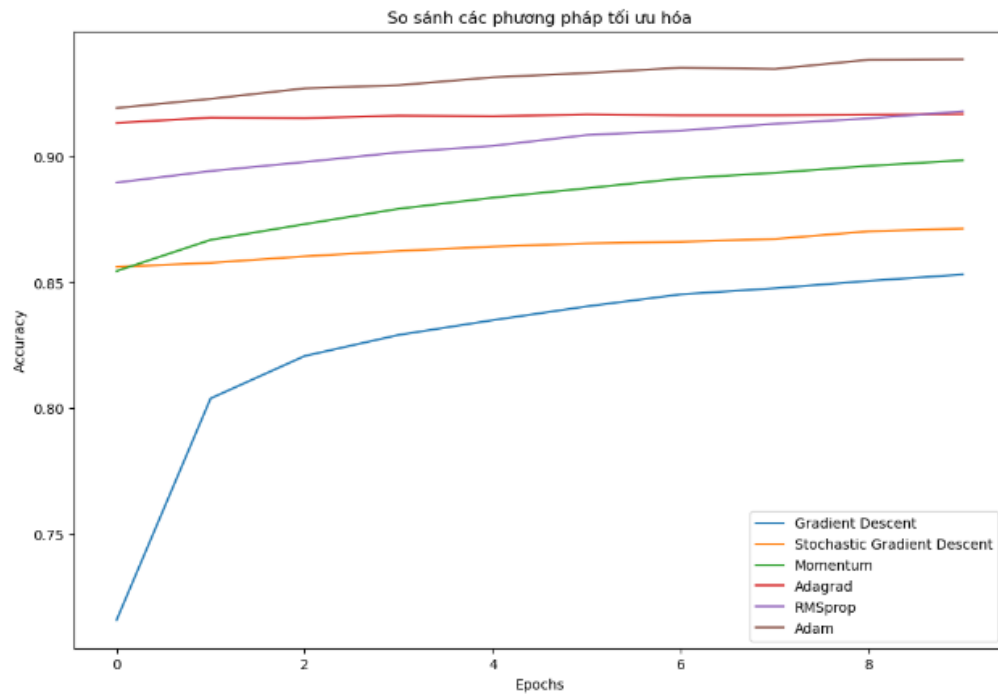
Hình 4. 2 Hình ảnh quá trình huấn luyện của mô hình

```

# Các phương pháp tối ưu hóa cần được so sánh
optimizers = {
    'Gradient Descent': tf.keras.optimizers.SGD(),
    'Stochastic Gradient Descent': tf.keras.optimizers.SGD(momentum=0.0, nesterov=False),
    'Momentum': tf.keras.optimizers.SGD(momentum=0.9, nesterov=False),
    'Adagrad': tf.keras.optimizers.Adagrad(),
    'RMSprop': tf.keras.optimizers.RMSprop(),
    'Adam': tf.keras.optimizers.Adam()
}

```

Hình 4. 3 Các phương pháp tối ưu



Hình 4. 4 Kết quả so sánh

Vậy theo kết quả thì phương pháp Adam là phương pháp có độ chính xác cao và có độ ổn định tốt.

```
class ContinualLearningModel:
    def __init__(self):
        self.model = LinearRegression()
        self.previous_data = None

    def train(self, X_train, y_train):
        # Áp dụng continual Learning
        if self.previous_data is not None:
            X_train = np.concatenate([self.previous_data['X'], X_train])
            y_train = np.concatenate([self.previous_data['y'], y_train])

        self.model.fit(X_train, y_train)

    def test_production(self, X_test, y_test):
        # Kiểm thử sản xuất
        y_pred = self.model.predict(X_test)
        mse = mean_squared_error(y_test, y_pred)
        print(f"Mean Squared Error on production data: {mse}")

    def update_previous_data(self, X, y):
        # Cập nhật dữ liệu trước đó sau mỗi lần huấn luyện
        self.previous_data = {'X': X, 'y': y}
```

Hình 4. 5 Mô phỏng ContinualLearning và test production

```
# Chia dữ liệu thành tập huấn luyện và tập kiểm thử
X_train_1, X_test_1, y_train_1, y_test_1 = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
# Tạo và huấn luyện mô hình
model = ContinualLearningModel()
model.train(X_train_1, y_train_1)
model.test_production(X_test_1, y_test_1)
```

Mean Squared Error on production data: 0.03711379440797686

Hình 4. 6 Huấn luyện mô hình lần thứ nhất

```
# Cập nhật dữ liệu và tiếp tục huấn luyện
X_train_2, X_test_2, y_train_2, y_test_2 = train_test_split(X, y, test_size=0.2, random_state=123)
model.update_previous_data(X_train_1, y_train_1)
model.train(X_train_2, y_train_2)
model.test_production(X_test_2, y_test_2)
```

Mean Squared Error on production data: 0.04789760484101265

Hình 4. 7 Huấn luyện mô hình lần thứ hai

Như trong đoạn mã trên ta thấy có thể lưu và tiếp tục update dữ liệu vào mô hình để tiếp tục huấn luyện theo kết quả trước đó.

TÀI LIỆU THAM KHẢO

1. What is Continuous Learning? Revolutionizing Machine Learning & Adaptability | DataCamp
2. Optimizer- Hiểu sâu về các thuật toán tối ưu (GD,SGD,Adam,..) (viblo.asia)
3. Thuật toán Gradient Descent cho Linear Regression | How Kteam