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 ĐẠT KHƯƠNG – 52100973**

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 ĐẠT KHƯƠNG - 52100973**

Lớp **: 21050401**

Khoá **: 25**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023

LỜI CẢM ƠN

Lời đầu tiên em xin gửi lời cám ơn đến toàn thể thầy cô giảng viên, Trường Đại học Tôn Đức Thắng đã tạo điều kiệu cho chúng em có điều kiện học tập tốt nhất.

Tiếp theo em xin gửi lời cám ơn sâu sắc đến thầy Lê Anh Cường, là giảng viên đảm nhận giảng dạy tụi em môn học Nhập môn học máy. Đây là một môn học rất thú và cùng với các giảng dạy rất dễ hiểu của thầy nên em đã tiếp thu được rất nhiều kiến thức bổ ích về những lý thuyết cơ bản của học máy, xây dựng được các mô hình, …

Lời cuối em xin chúc thầy luôn có được sức khỏe và ngày càng thành công hơn nữa trong sự nghiệp trồng người.

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 17 tháng 12 năm 2023

Tác giả

(ký tên và ghi rõ họ tên)

Khương

Nguyễn Đạt Khương

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

Bài báo cáo này để em trình bày về các nội dung trong phần tìm hiểu riêng trong báo cáo cuối kỳ môn Nhập môn học máy bao gồm nghiên cứu và so sánh các phương pháp Optimizer trong huấn luyện mô hình học máy, tìm hiểu về Continual Learning và Test Production khi xây dựng một giải pháp học máy

MỤC LỤC

DANH MỤC CÁC CHỮ VIẾT TẮT

1. GD: Gradient Descent
2. SGD: Stochastic Gradient Descent

DANH SÁCH CÁC BẢNG BIỂU, HÌNH VẼ

PHẦN 1: CÁC PHƯƠNG PHÁP OPTIMIZER

I. Tổng quan về Optimizer:

Trong lĩnh vực học máy hiện đại, việc áp dụng các phương pháp tối ưu hóa (Optimizer) và học liên tục (Continual Learning) đóng vai trò quan trọng trong quá trình huấn luyện mô hình. Đặc biệt, với sự phức tạp ngày càng tăng của dữ liệu hiện nay, việc tối ưu hóa mô hình để đạt được hiệu suất cao là không thể phủ nhận.

Việc sử dụng các phương pháp Optimizer và Continual Learning mang lại nhiều lợi ích. Đầu tiên, nó tăng cường hiệu suất của mô hình, cho phép mở rộng khả năng ứng dụng của nó. Thứ hai, nó gia tăng khả năng thích ứng của mô hình đối với các loại dữ liệu đa dạng. Thứ ba, thông qua quá trình tổng quát hóa dữ liệu, mô hình được đào tạo có khả năng duy trì hiệu suất cao trên dữ liệu mới mà không làm ảnh hưởng đến khả năng xử lý dữ liệu cũ.

Sự kết hợp linh hoạt giữa các phương pháp này chính là chìa khóa để xây dựng một mô hình học máy xuất sắc, có khả năng ứng dụng rộng rãi trong nhiều tình huống và ngữ cảnh khác nhau.

II. Mục tiêu của Optimizer:

Mục tiêu chính của Optimizer trong học máy là tối ưu hóa quá trình huấn luyện mô hình bằng cách điều chỉnh các tham số của mô hình để giảm thiểu giá trị của hàm mất mát. Optimizer hoạt động dựa trên gradient của hàm mất mát theo các tham số của mô hình, với mục tiêu là đưa mô hình đến một điểm cực tiểu cục hoặc cực tiểu toàn cục của hàm mất mát.

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

3.1. Các phương pháp Optimizer phổ biến:

3.1.1. Gradient Descent

Thông thường mục đích của các bài toán tối ưu hóa là ta sẽ tìm giá trị nhỏ nhất của hàm số nào đó mà hàm số đó đạt giá trị nhỏ nhất khi đạo hàm bằng 0 tuy nhiên không phải lúc nào chúng ta cũng có thể tìm được đạo hàm của bài toán vì các hàm số nhiều biến và rất phức tạp hoặc thậm chí làm không thể để tìm được đạo hàm.

Vậy nên thay vào đó, trong bài toán áp dụng Gradient Descent ta sẽ tìm điểm gần điểm cực tiểu nhất và xem đó là giá trị cần tìm

- Tối ưu: GD tối ưu hàm mất mát thường được kí hiệu là J(0), trong đó 0 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.

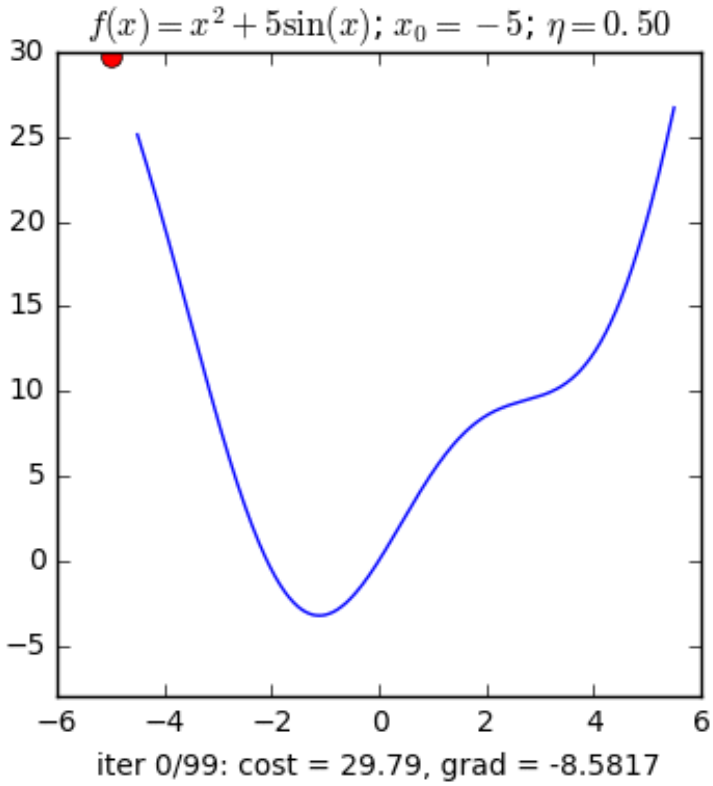
a) Công thức:

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

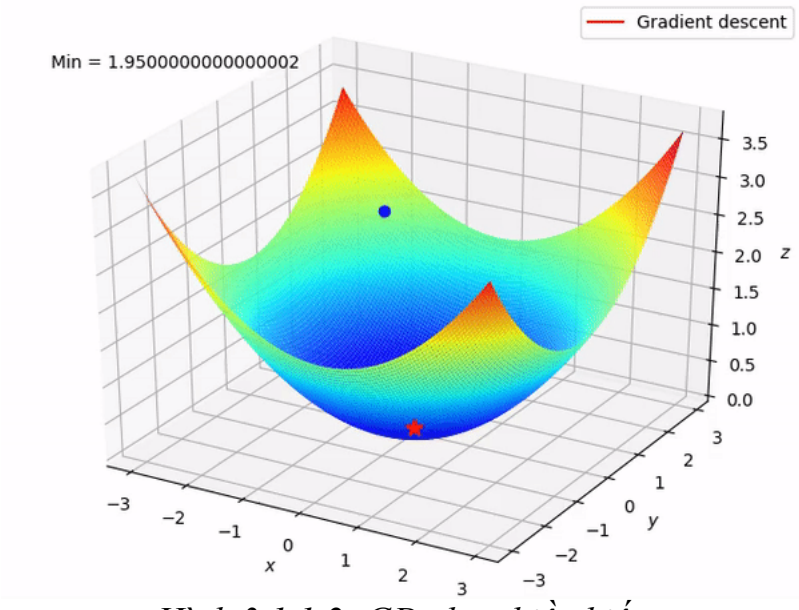
Trong đó, GD có các tham số:

- 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 trũng thấp nhất của hàm $J(\theta)$ vì qua mỗi vòng lặp $J(\theta)$ sẽ càng chính xác.



Hình 3.1.1.1: GD cho hàm 1 biến



Hình 3.1.1.2: GD cho nhiều biến

b) Ưu điểm:

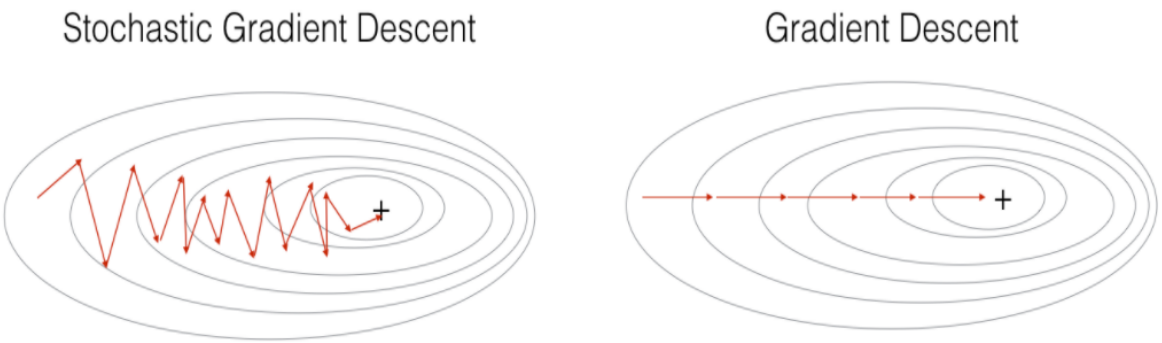
- Cơ bản dễ hiểu, giải quyết được vấn đề tối ưu hóa bằng cách cập nhật trọng số sau mỗi vòng lặp.

c) Nhược điểm:

- Phụ thuộc vào nghiệm khởi tạo ban đầu và learning rate.Learning rate lớn sẽ làm cho thuật toán kho hiệu quả.
- Ví dụ: một bài toán có 2 globlal minimum thì đầu ra sẽ là 2 kết quả khác nhau.
- Tốc độ học quá lớn sẽ gây ra hiện tượng không hội tụ (kết quả sẽ chỉ quanh quẩn quanh giá trị đích vì bước nhảy quá lớn), tốc độ học quá nhỏ thì ảnh. hưởng đến tốc độ training

3.1.2. Stochastic Gradient Descent (SGD):

Là một biến thể của Gradient Descent, thay vì ta thực hiện cập nhật trọng số trên mỗi vòng lặp như trong GD thì SGD sẽ tiến hành cập nhật lại trọng số đó trong từng biến.



Hình 3.1.2: So sánh giữa SGD và GD

- Sự khác biệt giữa hai thuật toán được thể hiện trong hình, dễ thấy được rằng thuật toán SGD có hình dạng ziczac vì 1 điểm không thể đại diện được cho toàn bộ bộ dữ liệu

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:

- Được áp dụng cho các cơ sở dữ liệu lớn mà GD không làm được

c. Nhược điểm:

- Vẫn còn các vấn đề tồn động của GD như learning rate và điểm khởi đầu, điều này yêu cầu kết hợp với các thujaatj toan khác như Momentum, AdaGrad, …

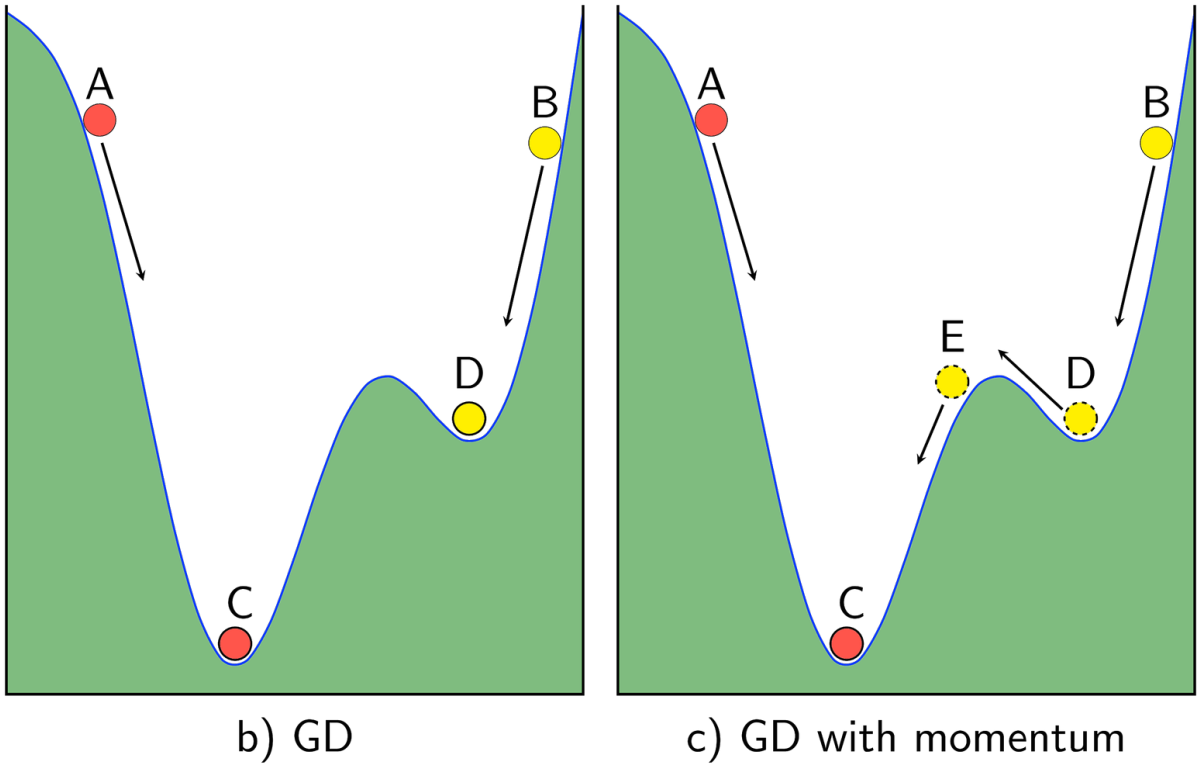
3.1.3. Momentum

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:**

Hiểu một các đơn giản, nếu ta xem thuật toán Gradient Descent của ta là một viên bi (A và B trong hình) và các điểm cực tiểu là các vực (C và D) là các nơi mà thuật toán (hay viên bi) cần chạm đến. Như đã trình bài ở trên, nếu ta đơn giản chỉ áp dụng GD thì viên bi B sẽ chỉ lăn đến điểm D(điểm cực tiểu cục bộ - local minimum) chứ không phải điểm C - global minimum) mà chúng ta mong

muốn.



Hình 3.1.3.1: GD có và không có momentum

Để giải quyết vấn đề trên, ta cấp cho viên bi - thuật toán một lực để nó có thể vượt qua điểm cực tiểu cục bộ đó - lý do GD with momentum ra đời với momentum là lực đẩy đó.

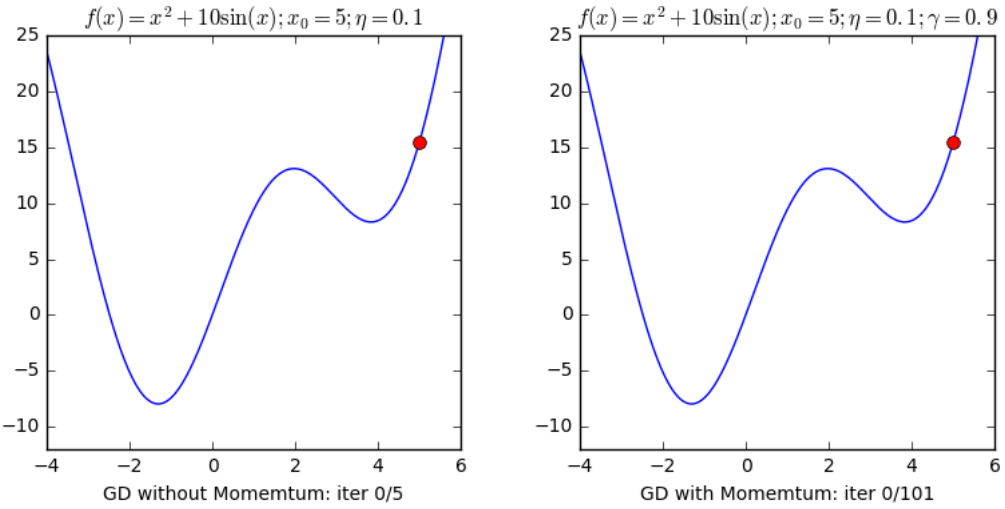
b. **Công thức:**

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

Trong đó:

- vt là vận tốc tại t
- θ_t là vector trọng số tại t
- η là learning rate

- $J(\theta_t)$ là hàm mất mát.



Hình 3.1.3.2: GD không có và có momentum

c. **Ưu điểm:**

- Giải quyết được vấn đề thuật toán Gradient Descent không tiến tới được Global Minimum mà chỉ dừng lại ở Local Minimum.

d. **Nhược điểm:**

- Dễ thấy ở hình trên, thuật toán GD with Momentum mất thêm khá nhiều vòng lặp dao động trước khi đạt được đến đích (điều này có thể hiển tương tự một viên bi còn có đà).

3.1.4. 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. Nguyên lý hoạt động:

- AdaGrad tích lũy bình phương của gradient cho mỗi tham số và sử dụng nó để điều chỉnh tỉ lệ học. Các tham số mà có gradient lớn hơn sẽ có tỉ

lệ học giảm nhanh chóng hơn, trong khi các tham số có gradient nhỏ sẽ có tỉ lệ học tăng lên. Điều này giúp đối phó tốt với vấn đề của các tham số có độ nhạy cao và tham số có độ nhạy thấp.

b. Công thức:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{G_t + \epsilon}} \odot g_t$$

Trong đó:

- $\theta_{t,i}$ là giá trị của tham số thứ i tại t
- η là learning rate, hằng số.
- $g_{t,i}$là gradient của tham số thứ i tại thời điểm t.
- $G_{t,(i,i)}$ là ma trận chéo mà mỗi phần tử trên đường chéo (i,i) là bình phương của đạo hàm vectơ tham số tại thời điểm t.
- $\in$ là tham số chống chia cho 0 (hay còn gọi là hệ số tránh lu

c. Ưu điểm:

- Tránh việc phải cập nhật LR một cách thủ công

d. Nhược điểm:

- Tổng bình phương sẽ biến thiên và lớn dần(tốc độ học sẽ giảm dần đến khi đạt cực tiểu)theo thời gian và có thể gây ra hiện tượng đóng băng thuật toán.

3.1.5. SMSProp

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.

Giải quyết được vấn đề đóng băng do LR giảm dần của Adagrad bằng cách chia tỉ lệ học cho trung bình bình phương của gradient

a. Công thức:

$$G_t = \gamma G_{t-1} + (1-\gamma) g_t^2$$
$$w_{t+1} = w_t - \frac{\eta}{\sqrt{G_t + \epsilon}} \cdot g_t$$

Trong đó:

- wt là giá trị của tham số tại thời điểm t
- gt là gradient tại thời điểm t
- Gt là tổng bình phương của các gradient trước đó cho tham số đó
- γ là hệ số giảm ảnh hưởng (decay rate hay hệ số giảm tỉ lệ học)
- η là tỉ lệ học
- ϵ là một hằng số nhỏ được thêm vào để tránh chia cho số không

- Công thức cập nhật trọng số:

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

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
- $\in$ là tham số chống chia cho 0

b. Ưu điểm:

- Giải quyết được vấn đề đóng băng như đã trình bày.

c. Nhược điểm:

- Kết quả trả về có thể chỉ là local minimun chứ không phải là global minimum

3.1.6. Adam

- Là sự kết hợp giữa Momentum và SMsprop.
- Để dễ hình dung, tương quan với việc xem Momentum là một viên bi thì ta sẽ xem Adam là một viên bi có lực ma sát rất lớn và có thể dễ dàng đến được global minimum và không mất quá nhiều vòng lặp.

a. Nguyên lý hoạt động:

- Việc thực hiện tối ưu hóa theo phương pháp Optimizer trải qua 4 giai đoạn:
  - Tính toán Gradient.
  - Hiệu chỉnh Bias của Gradient.
  - Hiệu chỉnh Scale của Gradient.
  - Hiệu chỉnh tỉ lệ học.

b. Công thức:

- Momentum:

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

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:

$$v_t = \beta_2 . v_{t-1} + (1 - \beta_2).(g^2{}_t)$$

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:

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

Trong đó:

- θ_t là vector trọng số tại t
- η là learning rate
- $\in$ là tham số chống chia cho 0

- v_t là vector RMSprop tại t
- m_t là vector vận tốc tại t

c. 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.

d. Nhược điểm:

- Nhạy cảm với nhiễu trong dữ liệu.
- Vấn đề overfitting khi quá hiệu quả.

3.2. So sánh và đánh giá:

Với từng ưu nhược điểm của các phương pháp tối ưu hóa nói trên, ta có thể đưa ra một nhận xét chung về các mô hình học máy nói trên:

- **Nguyên lý hoạt động:**
  - Gradient Descent: Di chuyển theo ngược hướng của gradient để giảm giá trị hàm mất mát.
  - SGD: Áp dụng gradient descent trên một hoặc một vài điểm dữ liệu ngẫu nhiên.
  - Momentum: Sử dụng mômentum để tích lũy động lượng và giảm tốc độ của quá trình hội tụ.
  - RMSProp và Adam: Kết hợp mômentum và trung bình động của bình phương gradient để điều chỉnh tỉ lệ học.
- **Thích ứng tỉ lệ học:**
  - RMSProp và Adam: Tự động điều chỉnh tỉ lệ học cho từng tham số dựa trên lịch sử của gradient.

- Momentum và GD: Sử dụng tỉ lệ học cố định, có thể cần điều chỉnh thủ công.

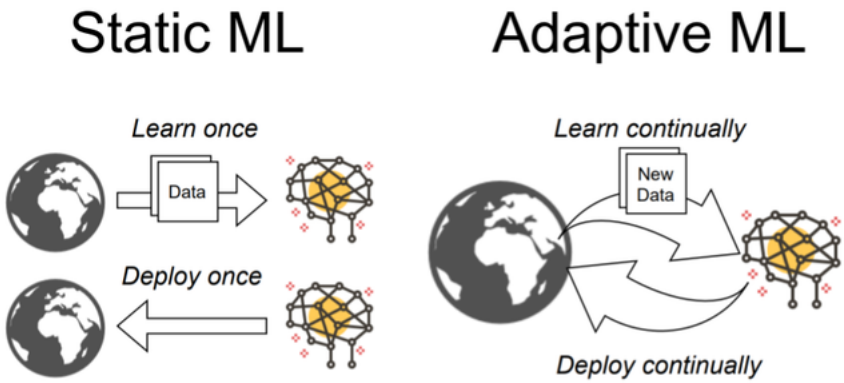
- **Độ nhạy đối với dữ liệu:**
  - Momentum và RMSProp: Giảm độ nhạy đối với nhiễu và biến động của gradient.
  - SGD và GD: Có thể bị ảnh hưởng bởi nhiễu, đặc biệt là trên dữ liệu nhỏ.
- **Hiệu quả trên dữ liệu độc lập:**
  - Adam thường hoạt động tốt trên nhiều mô hình tác vụ.
  - Momentum và RMSProp: Hiệu quả trên một số tác vụ nhất định.
- **Sự hội tụ:**
  - Adam: Thường hội tụ nhanh chóng trên nhiều tình huống.
  - SGD: Cần lựa chọn tỉ lệ học thích hợp để đảm bảo sự hội tụ.

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

4.1. Continual Learning:

a. Khái niệm:

Là một trong 4 định nghĩa Continuous Intergration (CI), Continous Delivery (CD), Continuos Training (CT) và Continuous Machine Learning (CML). 4 khái niệm này được sinh ra để giải quyết cập nhật thuật toán liên tục vì trên thực tế các bộ dữ liệu được cập nhật thường xuyên.



Hình 4.1.1: Mô hình Continual Learning

Hay còn được gọi là Incremental Learning and Lifelong Learning (học cả đời) là một khái niệm mà tại đó mô hình học máy được xây dựng để thực hiện một lượng lớn các yêu cầu tác vụ một cách liên tục mà không quên đi các kiến thức đã học được từ các tác vụ trước đó. Mô hình này có thể được xem là một thuật toán có khả năng thích ứng cao với các số lượng nhiệm vụ cần học không được định nghĩa trước.Cách thức xây dựng:

- Lưu trữ dữ liệu đào tạo mới khi mô hình nhận được dữ liệu đó.
- Khi có đủ dữ liệu mới, đánh giá độ chính xác của nó với mô hình đang xây dựng.
- Nếu thấy độ chính xác của mô hình giảm dần theo thời gian, sử dụng kết hợp bộ dữ liệu cũ và mới để tạo ra một mô hình học máy tốt hơn.

b. Qui trình của Continual Learning

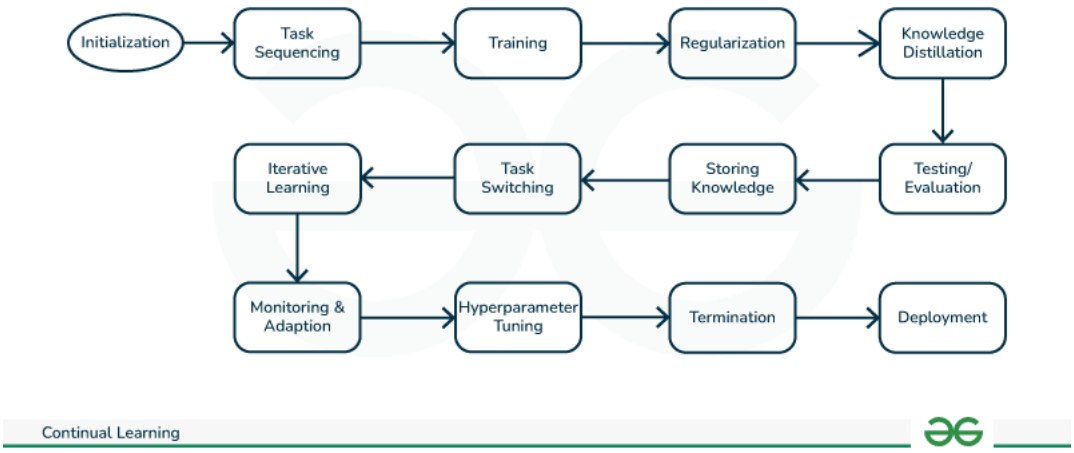
Quy trình của continual learning là sự phát triển của mô hình học máy cơ bản.

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.



Hình 4.1.2: Quy trình Continual Learning

c. Ưu điểm của Continual learning:

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.

- **Tính linh hoạt:** Mô hình có khả năng học từ dữ liệu mới mà không cần đào tạo lại toàn bộ mô hình. Điều này giúp tận dụng thông tin mới một cách linh hoạt và tiết kiệm tài nguyên đào tạo.
- **Khả năng mở rộng:** Continual Learning cho phép mô hình mở rộng khả năng của mình để xử lý dữ liệu mới và nhiệm vụ mới mà không cần bắt đầu từ đầu.
- **Bảo quản kiến thức cũ:** Mô hình được thiết lập để giữ lại kiến thức đã học từ dữ liệu cũ trong quá khứ, ngăn chặn hiện tượng quên mất thông tin quan trọng.
- **Tiết kiệm tài nguyên:** Thay vì đào tạo lại mô hình từ đầu, Continual Learning giúp tiết kiệm tài nguyên tính toán và thời gian đào tạo, đặc biệt quan trọng khi có lượng lớn dữ liệu và tài nguyên hạn chế.
- **Đảm bảo tính khái quát:** Continual Learning giúp mô hình duy trì khả năng tổng quát đối với dữ liệu mới mà không làm giảm hiệu suất trên dữ liệu cũ, đảm bảo tính tổng quát của mô hình.
- **Chống lão hóa mô hình:** Hạn chế hiện tượng lão hóa mô hình, tức là sự suy giảm hiệu suất khi mô hình phải xử lý dữ liệu mới.
- **Ứng dụng trong môi trường thực tế:** Thích ứng tốt với các ứng dụng thực tế, nơi mô hình cần liên tục cập nhật để đối mặt với sự biến động và thay đổi của môi trường.
- **Khả năng đối mặt nhiệm vụ đa dạng:** Continual Learning giúp mô hình đối mặt với nhiều loại nhiệm vụ và tình huống khác nhau mà không cần kiến thức trước đó.

d. Nhược điểm của continual learning

- Khi xác định sai vấn đề từ đầu, sẽ khó tạo ra được một chương trình học máy ứng dụng phục vụ đúng được yêu cầu của người dùng.
- Thu thập các dữ liệu cần thiết: một mô hình học máy liên tục yêu cầu một bộ chính xác và chất lượng cao.
- Duy trì mô hình: khi các dữ liệu thay đổi thì mô hình học có thể trở nên lỗi thời nên cần có sự can thiệp của con người, bảo trì, quản lý liên tục và điều chỉnh hướng đi để mang lại kết quả có ý nghĩa.

e. Ứng dụng của Continual learning

- **Hệ thống dự đoán và phân quyền liên tục:**
  - **Dự báo thị trường:** Đối với các mô hình dự đoán thị trường tài chính hoặc doanh nghiệp, continual learning giúp cập nhật mô hình với dữ liệu thị trường mới và biến động liên tục.
  - **Phân loại hình ảnh và video:** Trong các ứng dụng như giám sát an ninh, mô hình có thể học từ dữ liệu mới để nhận diện các đối tượng mới hoặc biến đổi trong môi trường.
- **Xử lý ngôn ngữ tự nhiên:**
  - **Chatbots thông minh:** Chatbots có thể học từ trải nghiệm mới và dữ liệu ngôn ngữ mới để cải thiện khả năng tương tác và hiểu người dùng.
  - **Phân loại văn bản:** Các hệ thống phân loại văn bản có thể liên tục cập nhật kiến thức về từ ngữ và ngữ cảnh mới.
- **Ô tô tự động và Robot:**

- **Hệ thống lái xe tự động:** Các mô hình có thể liên tục học từ dữ liệu lái xe mới để cải thiện khả năng đối phó với các tình huống giao thông mới và biến động đường.

- **Quản lý dữ liệu và an toàn thông tin:**
  - **Phát hiện đối tượng độc hại:** Mô hình có thể học từ các mẫu mới để phát hiện và ngăn chặn các hành vi độc hại hoặc tấn công mạng mới.
  - **Quản lý rủi ro an toàn thông tin:** Các hệ thống có thể liên tục cập nhật kiến thức về các mô hình an toàn và biện pháp bảo mật mới.

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.

"Test production" trong lĩnh vực machine learning (ML) thường được hiểu là quá trình kiểm thử mô hình ML sau khi nó đã được triển khai vào môi trường sản xuất. Đây là một bước quan trọng để đảm bảo rằng mô hình hoạt động đúng đắn và hiệu quả trong môi trường thực tế. Test Product đối với Machine learning thường được chia làm 2 phần chính là Testing và Evaluation.

b. So sánh Evaluation và Testing Production

a. Evaluation Production:

- Hay còn được gọi là đánh giá mô hình với các đặc điểm như sau:
    - Tập trung vào hiệu suất tổng thể của chương mô hình dựa vào các số liệu hay đường cong hiệu suất hoặc các ví dụ về dự đoán không chính xác của mô hình.
    - Đánh giá mô hình học máy là một cách rất tốt để theo dõi kết quả của các phiên bản mô hình học máy khác nhau.

b. Testing Production:

- Khác với evaluation (đánh giá) thì testing (kiểm thử) không chỉ đánh giá hiệu quả mô hình mà còn phải đảm bảo được là các bộ phận tổng hợp của một mô hình học máy hoạt động hiệu quả để đạt được kết quả mong muốn bằng việc chỉ ra các sai sót trong mã, dữ liệu và mô hình để sửa chữa.
- Các vấn đề khi kiểm thử mô hình học máy:
    - **Thiếu ổn định:** kết quả mô hình không xác định, nhiều thuật toán dựa và sự ngẫu nhiên và tạo ra mô hình mới không giống mô hình ban đầu sau khi đào tạo lại.
    - **Tính tổng quát hóa:** các mô hình học máy phải nhất quán trong nhiều hoàn cảnh đào tạo khác nhau.
    - **Ý tưởng không rõ ràng về phạm vi:** không có một quy định cụ thể về phạm vi thử nghiệm của một mô hình học máy, bên cạnh đó, việc đánh giá một mô hình học máy không dựa trên những dòng code mà dựa vào bộ dữ liệu đầu vào và ra của mô hình.
    - **Nhu cầu về nhân lực:** việc cập nhật liên tục của mô hình học máy đòi hỏi tiêu tốn rất nhiều nhân lực và tài nguyên để kiểm thử.

TÀI LIỆU THAM KHẢO

1. Optimizer – Hiểu sâu về thuật toán tối ưu.
2. Optimizer-Deep Learning
3. 10 Famous Machine Learning Optimizer
4. Introduction to Continual Learning
5. What is Continual Machine Learning