**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG KHOA CÔNG NGHỆ THÔNG TIN**



# BÀI ĐỒ ÁN CUỐI KÌ MÔN NHẬP MÔN HỌC MÁY

**Dự án cuối kì**

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

*Người thực hiện*: **PHẠM QUANG ĐỨC – 52100880**

## Lớp : 21050301

## THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG KHOA CÔNG NGHỆ THÔNG TIN**



# BÀI ĐỒ ÁN CUỐI KÌ MÔN NHẬP MÔN HỌC MÁY

**Dự án cuối kì**

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

*Người thực hiện*: **PHẠM QUANG ĐỨC – 52100880**

## Lớp : 21050301

## THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023

# LỜI CẢM ƠN

Em xin chân thành cảm ơn thầy Lê Anh Cường, người đã dành tâm huyết và thời gian để hướng dẫn chúng em trong quá trình thực hiện đồ án cuối kì. Sự chia sẻ kiến thức sâu rộng và tận tâm của thầy không chỉ giúp chúng em vượt qua những thách thức trong nghiên cứu mà còn là nguồn động viên lớn, làm cho hành trình này trở nên ý nghĩa và giáo dục hơn. Thầy không chỉ là những người hướng dẫn mà còn là những người đồng hành, người bạn quan trọng trong quá trình học tập. Chúng em xin bày tỏ lòng biết ơn sâu sắc và tri ân đến thầy, người đã góp phần làm nên thành công của công trình này. Cảm ơn thầy vì sự cống hiến và sự hỗ trợ vô song.

# 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 TS Nguyễn Văn A;. 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 tháng năm Tác giả*

*(ký tên và ghi rõ họ tên)*

*Phạm Quang Đức*

# 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

.

# MỤC LỤC

[LỜI CẢM ƠN i](#_bookmark0)

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN iv](#_bookmark1)

[TÓM TẮT v](#_bookmark2)

[MỤC LỤC 1](#_bookmark3)

[CHƯƠNG 1 2](#_bookmark4)

* 1. [Tìm hiểu và so sánh các phương pháp Optimizer trong huấn luyện mô hình học máy 2](#_bookmark5)
  2. [Tìm hiểu về Continual Learning và Test Production khi xây dựng một giải pháp học máy để giải quyết bài toán nào đó 10](#_bookmark5)

# CHƯƠNG 1

## Tìm hiểu và so sánh các phương pháp Optimizer trong huấn luyện mô hình học máy

## 1.1.1 Tổng quan về Optimizer

- Trong machine learning, optimizer (trình tối ưu) là một thành phần quan trọng trong việc huấn luyện mô hình. Mục tiêu của optimizer là tối ưu hoá hàm mất mát, tức là giảm thiểu độ lỗi giữa dự đoán của mô hình và giá trị thực tế của dữ liệu đào tạo. Chúng ta phải tìm một thuật toán để cải thiện weight và bias theo từng bước để tối ưu hoá model

- Một số hàm mất mát được định nghĩa để đo lường sự chênh lệch giữa giá trị dự đoán của mô hình và giá trị thực tế. Quá trình tối ưu hoá bằng cách điều chỉnh các tham số của mô hình dựa trên các giá trị mất mát này. Trong đó gradient của hàm mất mát đối với các tham số của mô hình được tính và sau đó được sử dụng để cập nhật các tham số đó.

## 1.1.2 Các thuật toán tối ưu

## 1.1.2.1 Gradient Descent (GD)

## - Gradient Descent là một thuật toán tối ưu hoá được dùng để điều chỉnh các tham số của mô hình học máy (machine learning) nhằm giảm thiểu hàm mất mát (loss function). Mục tiêu là điều chỉnh các tham số sao cho hàm mất mát đạt giá trị nhỏ nhất, tức là mô hình gần với dự đoán thực thế nhất. Đơn giản hơn là gradient descent là thuật toán tìm ra đường thẳng fit nhất từ bộ dữ liệu cho sẵn

## 

## - Điểm xanh lục là điểm local minimum (cực tiểu), và cũng là điểm làm cho giá trị hàm số đạt nhỏ nhất

## - Điểm local minimum là nghiệm của phương trình có đạo hàm bằng 0. Trong hầu hết các trường hợp, việc phương trình đạo hàm bằng 0 là bất khả thi. Do các điểm dữ liệu có chiều lớn, hoặc từ việc có quá nhiều điểm dữ liệu.

## - Hướng tiếp cận phổ biến xuất phát từ một điểm mà ta coi là gần với nghiệm của bài toán sau đó dùng các phép toán lặp để tiến gần các điểm cần tìm là các điểm có đạo hàm gần bằng 0

## 

## - Các bước thực hiện gradient descent

## + Bắt đầu với một bộ tham số ban đầu. Thường là các giá trị được chọn ngẫu nhiên hoặc giá trị được chọn một cách thông thường

## + Đạo hàm riêng của hàm mất mát của hàm tham số

## + Tham số được điều chỉnh theo hướng của gradient để giảm giá trị của hàm mất mát. Tốc độ điều chỉnh tham số gọi là learning rate

## + Lặp lại quá trình

## - Gradient Descent cho hàm nhiều biến

## + Ta có hàm giả thuyết

## 

## + Hàm chi phí của multivariate linear regression

## 

## + Có thể viết đơn giản thành

## 

## + Gradient của hàm chi phí như sau

## 

## + Có thể viết đơn giản hơn là

## 

## 

## 1.1.2.2 Stochastic Gradient Descent (SGD)

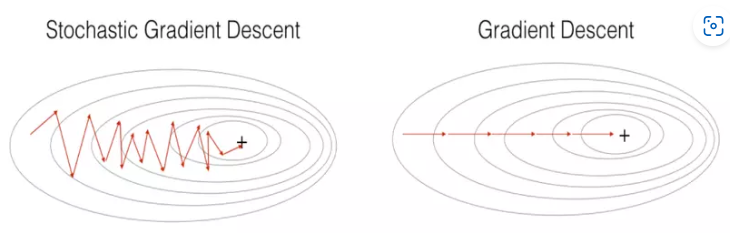
- Trong thuật toán, tại một thời điểm, ta chỉ tính đạo hàm của hàm mất mát dựa trên chỉ một điểm dữ liệu xi rồi cập nhật theta dựa trên đạo hàm này.

- Mỗi lần duyệt một lượt qua tất cả các điểm trên toàn bộ dữ liệu được gọi là một epoch. Với GD thông thường thì mỗi epoch ứng với một lần cập nhật theta. Với SGD thì mỗi epoch ứng với N lần cập nhật theta với N điểm dữ liệu

- Thứ tự lựa chọn điểm dữ liệu, sau mỗi epoch chúng ta cần shuffle thứ tự của các dữ liệu để đảm bảo tính ngẫu nhiêu.

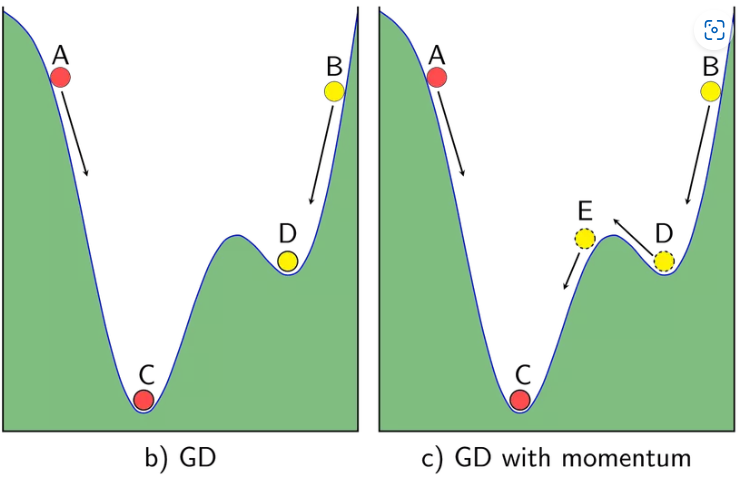


- Trong đó  là hàm mất mát với chỉ mọt cặp dữ liệu (input, label)



**1.1.2.3 Momentum**

- Để khắc phục hạn chế của thuật toán Gradient Descent người ta dùng gradient descent with momentum



- Khi điểm ta xét di chuyển về điểm local minimum (D) nhưng ta không muốn nó dừng lại ở điểm local mà sẽ lăn xuống xuống điểm global (C)

- Trong GD, chúng ta cần tính lượng thay đổi ở thời điểm t để cập nhật vị trí mới cho nghiệm (tức hòn bi). Nếu chúng ta coi đại lượng này như vận tốc trong vật lý, vị trí mới của hòn bi sẽ là . Một cách đơn giản nhất, ta có thể cộng 2 đại lượng này lại

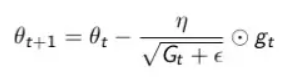


-  γ được chọn là một giá trị khoảng 0.9, vt là vận tốc tại thời điểm trước đó, ∇θJ(θ) chính là độ dốc của điểm trước đó. Sau đó vị trí mới của hòn bi được xác định như sau:

θ=θ−vt

**1.1.2.4 Adagrad**

**-** Adagrad là một thuận toán để tối ưu hoá dựa trên gradient descent, thuật toán này giúp điều chỉnh tốc độ học ứng với các chức năng khác nhau, thực hiện các cập nhật nhỏ hơn (tức là tốc độ học thấp) cho các tham số liên quan đến các tính năng thường xuyên xuất hiện và các cập nhật lớn hơn (tức là tốc độ học cao) cho các tham số liên quan đến các tính năng không thường xuyên. Adagrad thích hợp để xử lý các dữ liệu thưa thớt



- Trong đó:

+ n là hằng số

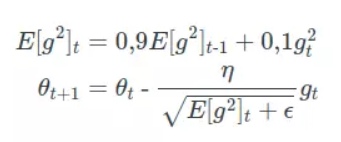
+ gt là gradient tại thời điểm t

+ ϵ là hệ số tránh lỗi

+ G là ma trận chéo mà mỗi phần tử trên đường chéo (i,i) là bình phương của đạo hàm vector tham số tại thời điểm t

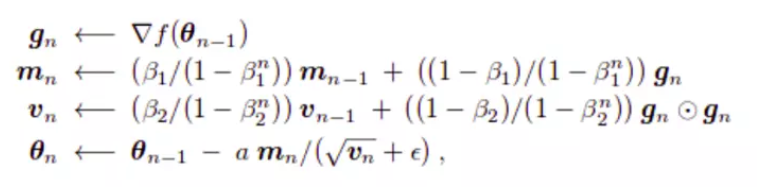
**1.1.2.5 RMSPROP**

**-** Thuật toán RMSProp là điều chỉnh tỷ lệ học cho mỗi tham số dựa trên trung bình độ lớn của các gradient descent gần đây. Tỷ lệ học thích ứng này giúp tăng tốc quá trình tối ưu hoá. Thuật toán đạt được điều này bằng cách chia tỷ lệ học cho mỗi tham số cho căn bậc 2 của trung bình mô phỏng giảm dần theo mũ độ lớn gradient bình phương

****

**1.1.2.6 Adam**

**-** Adam là thuật toán tối ưa hoá phổ biến trong huấn luyện mạng nơ-ron để cải thiện hiệu quả và tốc độ hội tụ. Nó là sự kết hợp của Momentum và RMSprop. Nếu giải thích theo hiện tượng vật lí thì Momentum giống như 1 quả cầu lao xuống dốc, còn Adam như một quả cầu nặng có ma sát, vì vậy nó dễ dàng vượt qua local minimum tới global minimum và khi tới global minimum nó không mất nhiều thời gian dao động qua lại quanh đích vì nó có ma sát nên dễ dừng lại hơn

****

**1.1.2.7 So sánh các Optimizer**

|  |  |  |
| --- | --- | --- |
| Optimizer | Ưu điểm | Nhược điểm |
| Gradient Descent | - Cơ bản, dễ hiểu. giải quyết được vấn đề tối ưu model nẻual network bằng cách cập nhật trọng số sau mỗi vòng lăoj | - Vì nó đơn giản nên còn nhiều hạn chế như phụ thuộc vào nghiệm khởi tạo ban đầu và learning rate  - Tốc độ học quá lớn sẽ khiến cho thuật toán không hội tụ, quanh quẩn bên đích vì bước nhảy quá lớn. Tốc độ học nhỏ cũng ảnh hưởng đến tốc độ training |
| Stochastic Gradient Descent (SGD) | - Giải quyết được đố với cơ sở dữ liệu lớn mà GD không làm được. Thuật toán tối ưu hiện nay vẫn hay được sử dụng | - Thuật toán chưa giải quyết được nhược điểm lớn của gradient descent (learning rate, điểm dữ liệu ban đầu). |
| Momentum | - Giải quyết được vấn đề GD không tiến tới được điểm global minimum mà chỉ dừng lại ở local minimum | - Tuy momentum giúp điểm vượt dốc đến điểm đích. Tuy nhiên khi tới gần đích, nó mất khá nhiều thời gian dao động qua lại trước khi dừng hẳn. Điều này được giải thích vì viên bi có đà |
| Adagrad | - Tránh việc điều chỉnh learning rate bằng tay, chỉ để tốc độ học default là 0.01 thì thuật toán sẽ tự động điều chỉnh | - Yếu điểm của Adagrad là tổng bình phương biến thiên sẽ lớn dần theo thời gian cho đến khi nó làm tốc độ cực kì nhỏ, làm việc training trở lên đóng băng |
| RMSprop | - Giải quyết vấn đề tốc độ học giảm dần của Adagrad | - Có thể cho kết quả nghiệm là local minimum chứ không đạt được global minimum như Momentum. |
| Adam | - Tính ổn định và hiệu quả  - Thích ứng với tốc độ học  - Không yêu cần tuning tham số phức tạp | - Adam cần lưu trữ thêm thông tin cho mỗi tham số để theo dõi bình phương  - Khả năng tối ưu hoá trên một bài toán nhất định  - Không đảm bảo sự hội tụ |

## Tìm hiểu về Continual Learning và Test Production để xây dựng một giải pháp học máy để giải quyết một bài toán nào đó

## 1.2.1 Continual Learning

## - Khái niệm: Học liên tục là mô hình hiện đại của machine learning, tạo ra các mô hình học máy có thể tăng và biến đổi liên tục. Học liên tục cho phép các mô hình phù hợp với thời gian, thu thập số liệu thống kê và năng lực mới không xoá bỏ những dữ liệu cũ trng quá khứ.

## - Các đặc điểm chính của continual learning bao gồm

## + Khả năng học liên tục: Mô hình có khả năng học từ dữ liệu mới khi nó được giới thiệu mà không quên hoặc làm giảm hiệu suất trên các dữ liệu cũ

## + Khả năng học đa tác vụ: Mô hình có khả năng giải quyết nhiều tác vụ khác nhau, khả năng học từ một tác vụ có thể hỗ trợ cho việc giải quyết các tác vụ trong tương lai

## + Quản lý vấn đề cánh tranh: Giải quyết các vấn đề của việc học một tác vụ mới có thể làm mất thông tin đã học của tác vụ cũ để duy trì hiệu suất trên tất cả các tác vụ đã học

## + Tự động điều chỉnh kiến thức cũ: Mô hình có khả năng tự động điều chỉnh kiến thức cũ, chẳng hạn như sự tinh chỉnh trọng số để làm cho kiến thức cũ vẫn hữu ích trong ngữ cảnh của tác vụ mới

## - Ưu điểm:

## + Khả năng áp dụng linh hoạt: Mô hình được huấn luyện bằng continual learning có khả năng chuyển giao kiến thức từ tác vụ cũ sang tác vụ mới. Giúp tận dụng thông tin học được và áp dụng nó vào ngữ cảnh mới.

## + Tiết kiệm tài nguyên: Giảm thiều việc phải huấn luyện mô hình lại từ đầu khi có dữ liệu mới học tác vụ mới

## + Thích nghi và bền bỉ: Thích nghi với môi trường thay đổi và giữ lại kiến thức quan trọng từ các tác vụ đã giải quyết trước đó

## - Nhược điểm:

## + Catastrophic Forgetting: Khi mô hình quên mất thông tin về tác vụ cũ khi tập trung vào tác vụ mới. Điều này có thể suy giảm hiệu suất trên các tác vụ cũ

## + Quản lý interference: Xảy ra khi thông tin từ một tác vụ ảnh hưởng đến khả năng học của mô hình trên tác vụ khác

## +Yêu cầu lưu trữ và tính toán cao: Đòi hỏi lưu trữ lịch sử huấn luyện lơn và tính toán phức tạp hơn so với việc đào tạo mô hình một lần

## 1.2.2 Test Production

## - Test Production trong machine learning thường để cập đến quá trình triển khai mô hình đã được huấn luyện vào môi trường sản xuất, nơi nó sẽ xử lý dữ liệu thực tế và đưa ra dự đoán hoặc quyết định dựa trên thông tin mới

## - Một số khía cạnh cụ thể về quá trình Test Production

## + Triển khai mô hình: Sau khi mô hình machine learning đã được huấn luyện thành công và đánh giá trên tập dữ liệu kiểm thử, quá trình sẽ triển khai mô hình vào môi trường sản xuất

## + Xử lý dữ liệu thực tế: Trong môi trường sản xuất, mô hình sẽ phải xử lý dữ liệu thực tế từ nguồn dữ liệu đầu vào của nó. Dữ liệu này có thể khác biệt so với dữ liệu mà mô hình đã được huấn luyện

## + Kiểm soát hiệu xuất: Sau khi triển khai, mô hình cần được kiểm soát hiệu suất để đảm bảo rằng nó đang đưa ra dự đoán chính xác trong môi trường thực tế

## + Quản lý đồng bộ: Liên quan đến quá trình đảm bảo rằng mô hình được duy trì và đồng bộ với sự thay đổi trong dữ liệu và yêu cầu kình doanh

## + Giám sát và tối ưu hoá liên tục: Liên quan đến việc thiết lập các hệ thống giám sát để theo dõi hiệu suất của mô hình trong thời gian thưucj và thực hiện các biện pháp tối ưu hoá liên tục khi cần thiết

## 