

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



BÀI LÀM CÁ NHÂN NHẬP MÔN HỌC MÁY

Người hướng dẫn: **GV LÊ ANH CƯỜNG**

Người thực hiện: **PHẠM TRƯỜNG GIANG – 51900793**

Lớp : 19050401

Khoá : 23

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**



BÀI LÀM CÁ NHÂN NHẬP MÔN HỌC MÁY

Người hướng dẫn: **GV LÊ ANH CƯỜNG**

Người thực hiện: **PHẠM TRƯỜNG GIANG – 51900793**

Lớp : 19050401

Khoá : 23

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023

LỜI CẢM ƠN

Trước tiên với tình cảm sâu sắc và chân thành nhất, cho phép em được bày tỏ lòng biết ơn đến tất cả các cá nhân và tổ chức đã tạo điều kiện hỗ trợ, giúp đỡ em trong suốt quá trình học tập và nghiên cứu. Trong suốt thời gian từ khi bắt đầu học tập tại trường đến nay, em đã nhận được rất nhiều sự quan tâm, giúp đỡ của quý Thầy Cô và bạn bè.

Với lòng biết ơn sâu sắc nhất, em xin gửi đến quý Thầy Cô ở Khoa Công nghệ thông tin trường Đại học Tôn Đức Thắng đã truyền đạt vốn kiến thức quý báu cho chúng em trong suốt thời gian học tập tại trường. Nhờ có những lời hướng dẫn, dạy bảo của các thầy cô nên đề tài nghiên cứu của em mới có thể hoàn thiện tốt đẹp.

Một lần nữa, em xin chân thành cảm ơn thầy/cô – người đã trực tiếp giúp đỡ, quan tâm, hướng dẫn em hoàn thành tốt bài báo cáo này trong thời gian qua. Trong quá trình làm bài nếu có sai sót mong thầy/cô bỏ qua cho em.

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

Tôi xin cam đoan đây là sản phẩm đồ án của riêng chúng tôi và được sự hướng dẫn của GV 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 tháng năm

Tác giả

(ký tên và ghi rõ họ tên)

Phạm Trường Giang

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

Trình bày tóm tắt vấn đề nghiên cứu, các hướng tiếp cận, cách giải quyết vấn đề và một số kết quả đạt được, những phát hiện cơ bản trong vòng 1 -2 trang.

MỤC LỤC

LỜI CẢM ƠN	i
PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN	iii
TÓM TẮT	iv
MỤC LỤC	1
DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ	3
CHƯƠNG 1 – CÁC PHƯƠNG PHÁP OPTIMIZER TRONG HUẤN LUYỆN MÔ HÌNH HỌC MÁY	4
1.1 Giới thiệu	4
1.2 Optimizer là gì?	4
1.3 Các phương pháp Optimizer	4
1.3.1 Gradient Descent (GD)	4
1.3.2 Stochastic Gradient Descent (SGD)	6
1.3.3 Adam	6
1.3.4 RMSprop	8
1.3.5 Momentum	8
1.4 So sánh các phương pháp	10
CHƯƠNG 2 – CONTINUAL LEARNING VÀ TEST PRODUCTION TRONG MACHINE LEARNING	11
2.1 Continual Learning - Lifelong Learning	11
2.2 Test Production	11

DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT

CÁC KÝ HIỆU

CÁC CHỮ VIẾT TẮT

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

DANH MỤC HÌNH

No table of figures entries found.

CHƯƠNG 1 – CÁC PHƯƠNG PHÁP OPTIMIZER TRONG HUẤN LUYỆN MÔ HÌNH HỌC MÁY

1.1 Giới thiệu

Nội dung ở phần này tìm hiểu, đánh giá và so sánh các phương pháp optimizer thông dụng trong quá trình huấn luyện mô hình học máy. Các optimizer đóng vai trò quan trọng trong việc tối ưu hóa trọng số của mô hình, ảnh hưởng trực tiếp đến khả năng học và hiệu suất của mô hình.

1.2 Optimizer là gì?

Khi nhắc đến Machine Learning thì không thể không biết về tối ưu hoá. Trước khi đi sâu vào vấn đề thì chúng ta cần hiểu thế nào là thuật toán tối ưu (optimizers), đây cũng là phần quan trọng trong Machine Learning vì bản chất các bài toán của nó đều qui về bài toán tối ưu một hàm mất mát. Về cơ bản, thuật toán tối ưu là cơ sở để xây dựng mô hình neural network với mục đích "học" được các features (hay pattern) của dữ liệu đầu vào, từ đó có thể tìm một cặp weights và bias phù hợp để tối ưu hóa model. Chúng ta phải tìm một thuật toán để cải thiện weight và bias theo từng bước, và đó là lý do các thuật toán optimizer ra đời.

1.3 Các phương pháp Optimizer

1.3.1 *Gradient Descent (GD)*

Phương pháp Gradient Descent (GD) là một thuật toán tối ưu hóa được sử dụng rộng rãi trong máy học và học máy để tìm giá trị nhỏ nhất của một hàm mất mát. Ý tưởng cơ bản của GD là điều chỉnh các tham số của mô hình (ví dụ: trọng số - weight) dựa trên đạo hàm của hàm mất mát theo từng tham số, với mục tiêu là giảm thiểu giá trị của hàm mất mát.

Quá trình thực thi Gradient Descent (GD):

1. Khởi tạo tham số.
2. Lặp lại các bước sau cho mỗi epoch hoặc đến khi đáp ứng điều kiện dừng:
 - Tính đạo hàm: Tính đạo hàm của hàm mất mát theo từng tham số. Đây là gradient của hàm mất mát.
 - Cập nhật tham số: Dùng gradient để điều chỉnh các tham số theo hướng giảm nhanh nhất của hàm mất mát. Công thức cập nhật thường là:

$$\theta^{(nextstep)} = \theta - \eta \nabla_{\theta}$$

- Trong đó, θ là vector tham số (ví dụ: vector trọng số); η là tốc độ học (learning rate) - một siêu tham số quyết định độ lớn của bước di chuyển; ∇_{θ} là gradient của hàm mất mát.

Việc chọn learning rate là quan trọng để đảm bảo thuật toán hội tụ một cách hiệu quả. Nếu learning rate quá lớn, thuật toán có thể không hội tụ hoặc có thể "nhảy qua" điểm tối ưu. Ngược lại, nếu learning rate quá nhỏ, thuật toán có thể hội tụ rất chậm.

Gradient Descent là một phương pháp quan trọng trong tối ưu hóa mô hình học máy và cần được điều chỉnh đúng cách để đạt được hiệu suất tối ưu. Nhiều biến thể của thuật toán, như SGD và Mini-Batch GD, được phát triển để giải quyết các thách thức liên quan đến tính toán và khả năng xử lý dữ liệu lớn.

1.3.2 *Stochastic Gradient Descent (SGD)*

Stochastic Gradient Descent (SGD) là một biến thể của Gradient Descent (GD) được thiết kế để giảm bớt tính toán so với phiên bản truyền thống. Trong SGD, thay vì cập nhật trọng số dựa trên toàn bộ tập dữ liệu (như trong Gradient Descent), chúng ta chỉ sử dụng một mẫu dữ liệu ngẫu nhiên (một điểm dữ liệu) để tính gradient và cập nhật trọng số.

Quá trình thực thi Stochastic Gradient Descent (SGD):

1. Khởi tạo tham số.
2. Lặp lại các bước sau cho mỗi epoch hoặc đến khi đáp ứng điều kiện dừng:
 - Chọn mẫu ngẫu nhiên: Chọn một mẫu dữ liệu ngẫu nhiên từ tập dữ liệu.
 - Tính Gradient: Tính đạo hàm của hàm mất mát đối với mẫu dữ liệu đã chọn.
 - Cập nhật tham số: Dùng gradient tính được để điều chỉnh tham số theo công thức.

SGD thường được sử dụng trong các mô hình học máy, đặc biệt là khi mô hình đối mặt với dữ liệu lớn và khi tính toán là một hạn chế. Nó được ứng dụng rộng rãi trong huấn luyện mô hình học máy sâu (deep learning) và các mô hình phức tạp khác. Các biến thể khác như Mini-Batch Gradient Descent là sự kết hợp giữa SGD và Gradient Descent truyền thống để cân bằng giữa tốc độ học và khả năng xử lý dữ liệu.

1.3.3 *Adam*

Adam là một phương pháp tối ưu hóa được thiết kế để cải thiện hiệu suất của Gradient Descent thông qua việc kết hợp các ưu điểm của các phương pháp khác nhau như Momentum và RMSProp. Tên gọi "Adam" là viết tắt của Adaptive Moment Estimation. Nếu giải thích theo hiện tượng vật lý thì Momentum giống như 1 quả cầu lao xuống dốc, còn Adam như 1 quả cầu rất 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.

Quá trình thực thi:

1. Khởi tạo tham số: Khởi tạo các tham số như trọng số và momentums cho việc tính toán đạo hàm bậc nhất và bậc hai.
2. Lặp lại các bước sau cho mỗi epoch hoặc đến khi đáp ứng điều kiện dừng:
 - Tính Gradient: Tính gradient của hàm mất mát đối với các tham số.
 - Tính Momentums: Tính toán momentums bậc nhất (tích lũy gradient) và bậc hai (tích lũy gradient bình phương). Momentums này được tính theo công thức giống với phương pháp Momentum và RMSProp.
 - Cập nhật tham số: Dùng momentums tính toán được để điều chỉnh tham số theo công thức:

$$\theta = \theta - \frac{\alpha}{\sqrt{v} + \epsilon} \cdot m$$

- Trong đó: θ là vector tham số (ví dụ: vector trọng số); α là learning rate, một siêu tham số quyết định độ lớn của bước di chuyển; m là momentums bậc nhất; v là momentums bậc hai; ϵ là một số nhỏ để tránh chia cho 0.

Adam thường được sử dụng trong các mô hình học máy sâu (deep learning) và các bài toán phức tạp khác. Sự kết hợp giữa tính toán đạo hàm bậc nhất và bậc hai giúp phương pháp này thích hợp cho nhiều loại hàm mất mát và có khả năng tối ưu hóa hiệu quả. Tuy nhiên, cũng có thể yêu cầu sự điều chỉnh kỹ thuật của các siêu tham số như learning rate.

1.3.4 RMSprop

RMSprop (Root Mean Square Propagation) là một phương pháp tối ưu hóa được thiết kế để cải thiện hiệu suất của Gradient Descent, đặc biệt là trong việc xử lý các vấn đề của các phương pháp truyền thống như SGD (Stochastic Gradient Descent). RMSprop giảm độ chậm trễ của việc cập nhật tham số bằng cách sử dụng thông tin về độ lớn của gradient. RMSprop giải quyết vấn đề tỷ lệ học giảm dần của Adagrad bằng cách chia tỷ lệ học cho trung bình của bình phương gradient.

Quá trình thực thi:

1. Khởi tạo tham số: Khởi tạo các tham số như trọng số và một giá trị ban đầu cho bậc hai trung bình của độ chệch (squared gradient moving average).
2. Lặp lại các bước sau cho mỗi epoch hoặc đến khi đáp ứng điều kiện dừng:
 - Tính Gradient: Tính gradient của hàm mất mát đối với các tham số.
 - Tính bậc 2 trung bình của độ lệch (Squared Gradient Moving Average).
 - Cập nhật tham số: Dùng độ lớn của gradient và bậc hai trung bình để điều chỉnh tham số theo công thức:

$$\theta = \theta - \frac{\alpha}{\sqrt{v} + \epsilon} \cdot \nabla J(\theta)$$

RMSprop thường được sử dụng trong huấn luyện mô hình học máy và đặc biệt phổ biến trong các mô hình học sâu. Cùng với Adam, RMSprop là một trong những phương pháp tối ưu hóa thông dụng được sử dụng để cải thiện hiệu suất của thuật toán Gradient Descent.

1.3.5 Momentum

Momentum là một phương pháp tối ưu hóa được thiết kế để giảm bớt độ dao động và gia tăng tốc độ hội tụ của thuật toán Gradient Descent. Ý tưởng chính là tích lũy "momentum" của gradient, giúp giảm độ dao động và thậm chí giúp bước di chuyển vượt qua các điểm cực tiểu cục bộ.

Quá trình thực thi:

1. Khởi tạo tham số: Khởi tạo các tham số như trọng số và một giá trị ban đầu cho momentums.
2. Lặp lại các bước sau cho mỗi epoch hoặc đến khi đáp ứng điều kiện dừng:
 - Tính Gradient: Tính gradient của hàm mất mát đối với các tham số.
 - Tính toán momentums bậc nhất (tích lũy gradient) theo công thức:

$$m = \beta m + (1 - \beta) \nabla J(\theta)$$

Trong đó: β là hệ số momentums; $\nabla J(\theta)$ là gradient của hàm mất mát.

- Cập nhật tham số: Dùng momentums tính toán được để điều chỉnh tham số theo công thức:

$$\theta = \theta - \alpha \cdot m$$

Trong đó: θ là vector tham số; α là learning rate; m là momentums bậc nhất.

Momentum thường được sử dụng trong huấn luyện mô hình học máy và là một phương pháp tối ưu hóa phổ biến. Nó có thể được kết hợp với các phương pháp khác như RMSprop và Adam để cải thiện hiệu suất của thuật toán Gradient Descent trong các bài toán học sâu và máy học.

1.4 So sánh các phương pháp

Phương pháp	Tính chất	Ưu điểm	Nhược điểm	Tính ứng dụng
Gradient Descent	Batch optimization.	Stabilizes training, hội tụ đến điểm tối ưu.	Cần nhiều tính toán, chậm với dữ liệu lớn.	Phổ biến trong các mô hình học máy.
Stochastic GD (SGD)	Stochastic optimization.	Phù hợp với dữ liệu lớn, giảm tính toán.	Khả năng dao động, hướng di chuyển nhiều.	Phổ biến trong học máy sâu, dữ liệu lớn.
Adam	Adaptive learning rate và momentums.	Tự điều chỉnh learning rate, hiệu suất tốt.	Cần điều chỉnh siêu tham số, tiêu tốn bộ nhớ.	Phổ biến trong học máy sâu, độ chính xác cao.
RMSprop	Adaptive learning rate với thông tin gradient.	Hiệu quả với dữ liệu không đồng nhất.	Phù hợp không tốt với một số bài toán.	Thường được sử dụng trong các mô hình sâu.
Momentum	Tích hợp momentums để giảm dao động.	Giảm độ dao động, vượt qua cực tiểu cục bộ.	Cần điều chỉnh hệ số momentum.	Phổ biến trong học máy sâu, giảm độ dao động.

CHƯƠNG 2 – CONTINUAL LEARNING VÀ TEST PRODUCTION TRONG MACHINE LEARNING

2.1 Continual Learning - Lifelong Learning

Continual Learning (CL) hay Lifelong Learning là một phương pháp học máy tập trung vào việc duy trì và mở rộng kiến thức của mô hình khi có sự thay đổi trong dữ liệu đầu vào. Mục tiêu của Continual Learning là giữ cho mô hình giữ lại kiến thức đã học khi giải quyết các bài toán mới và không quên thông tin dữ liệu cũ đã học được. Điều này phản ánh khả năng của hệ thống học máy để liên tục mở rộng kiến thức và năng lực của nó theo thời gian.

Tuy nhiên, nó cũng đối mặt với nhiều thách thức do tính chất phức tạp của dữ liệu, đặc biệt là khi đối mặt với dữ liệu mới. Catastrophic Forgetting là hiện tượng phổ biến, mô hình quên mất kiến thức đã học trước đó khi đối mặt với dữ liệu mới.

Để đối mặt với những thách thức có thể xảy ra, có nhiều kỹ thuật và phương pháp như Elastic Weight Consolidation (EWC), Progressive Neural Networks (PNN), và các mô hình Memory-Augmented Networks đã được phát triển để giúp mô hình học và giữ lại kiến thức liên tục.

2.2 Test Production

Test Production là quá trình phát triển và triển khai các bài kiểm tra để đánh giá hiệu suất của mô hình học máy. Trong quá trình xây dựng giải pháp học máy, việc kiểm thử là bước quan trọng để đảm bảo rằng mô hình không chỉ hoạt động tốt trên dữ liệu huấn luyện mà còn có khả năng tổng quát hóa đối với dữ liệu mới.

Quy trình kiểm thử:

1. Chuẩn bị dữ liệu: Chuẩn bị một tập dữ liệu kiểm thử độc lập với tập dữ liệu huấn luyện để đánh giá khả năng tổng quát hóa của mô hình.

2. Chọn Thước Đo Hiệu Suất (Metric Selection): Xác định các thước đo hiệu suất phù hợp với bài toán cụ thể, ví dụ như accuracy, precision, recall, F1 score, hay các thước đo khác tùy thuộc vào yêu cầu của ứng dụng.
3. Kiểm Thử Hiệu Suất (Performance Testing): Thực hiện kiểm thử trên tập dữ liệu kiểm thử sử dụng các thước đo đã chọn để đánh giá hiệu suất của mô hình.
4. Đánh Giá và Tối Ưu Hóa: Dựa trên kết quả kiểm thử, đánh giá hiệu suất của mô hình và thực hiện các điều chỉnh hoặc tối ưu hóa nếu cần thiết.

TÀI LIỆU THAM KHẢO

Tiếng Việt

1. [Continual Learning | Papers With Code](#)
2. [Machine Learning cơ bản \(machinelearningcoban.com\)](#)
3. [Gradient Descent | Quy's blog \(ndquy.github.io\)](#)
4. [Thuật toán Gradient Descent | TopDev](#)
5. [Optimizer- Hiểu sâu về các thuật toán tối ưu \(GD,SGD,Adam,...\) \(viblo.asia\)](#)

PHỤ LỤC

Phần này bao gồm những nội dung cần thiết nhằm minh họa hoặc hỗ trợ cho nội dung luận văn như số liệu, biểu mẫu, tranh ảnh. . . . nếu sử dụng những câu trả lời cho một *bảng câu hỏi thì bảng câu hỏi mẫu này phải được đưa vào phần Phụ lục ở dạng nguyên bản* đã dùng để điều tra, thăm dò ý kiến; **không được tóm tắt hoặc sửa đổi**. Các tính toán mẫu trình bày tóm tắt trong các biểu mẫu cũng cần nêu trong Phụ lục của luận văn. Phụ lục không được dày hơn phần chính của luận văn.