TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CUỐI KÌ MÔN NHẬP MÔN HỌC MÁY**

**BÁO CÁO CUỐI KÌ**

*Người hướng dẫn*: **GV. LÊ ANH CƯỜNG**

*Người thực hiện*: **PHẠM KHÁNH LINH - 52100908**

**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 KÌ MÔN NHẬP MÔN HỌC MÁY**

**BÁO CÁO CUỐI KÌ**

*Người hướng dẫn*: **GV. LÊ ANH CƯỜNG**

*Người thực hiện*: **PHẠM KHÁNH LINH - 52100908**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023

LỜI CẢM ƠN

Lời nói đầu tiên, em xin phép được gửi lời cảm ơn chân thành nhất đến Khoa Công Nghệ Thông Tin đã trang bị bộ môn “Nhập môn học máy” vào chương trình giảng dạy. Đó là môn học vô cùng cần thiết và hữu ích cho sinh viên chúng em, nhờ đó bộ môn đã tạo tiền đề tư duy và chính là hành trang cho những bộ môn tiếp theo cũng như là kiến thức bắt buộc cho tương lai chúng em sau này. Đặc biệt, em xin gửi lời cảm ơn sâu sắc đến ***giảng viên bộ môn - thầy Lê Anh Cường*** người đã hướng dẫn em trong môn học này. Thầy không chỉ là là người hướng dẫn em vê mặt lý thết mà còn tạo điều kiện cho em áp dụng kiens thức thông qua đồ án môn học. Nhờ sự hỗ trợ và giảng dạy tận tâm của thầy, em đã có cơ hội rèn luyện kỹ năng và ứng dụng kiến thức vào thực tế.

Qua những buổi học, em đã được truyền dạy kiến thức cơ bản để có thể tiến hành làm bài đồ án này, nhưng cũng do kiến thức và kinh nghiệm còn hạn hẹp nên đồ án của em cũng không thể tránh khỏi những thiếu sót. Em rất mong nhận được sự góp ý nhiều hơn từ thầy/cô giảng viên bộ môn.

***Chúng em xin chân thành cảm ơn!***

**ĐỒ ÁN ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi và được sự hướng dẫn khoa học của Thầy 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 luận văn 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 luận văn 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 08 tháng 12 năm 2023*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Phạm Khánh Linh*

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

Đây là bài đồ án cuối kỳ môn học nhập môn học máy, gồm có những nội dung sau:

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

[THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023 2](#_Toc154221397)

[LỜI CẢM ƠN i](#_Toc154221398)

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN ii](#_Toc154221399)

[TÓM TẮT iv](#_Toc154221400)

[MỤC LỤC 1](#_Toc154221401)

[DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ 2](#_Toc154221402)

[CHƯƠNG 1 – TÌM HIỂU, SO SÁNH CÁC PHƯƠNG PHÁP OPTIMIZER TRONG HUẤN LUYỆN MÔ HÌNH HỌC MÁY 3](#_Toc154221403)

[1.1 Mục tiêu của tối ưu hóa mô hình học máy 3](#_Toc154221404)

[1.2 Phương pháp Gradient Descent (GD) 3](#_Toc154221405)

[1.3 Phương pháp Stochastic Gradient Descent (SGD) 4](#_Toc154221406)

[1.4 Gradient Descent với Momentum 6](#_Toc154221407)

[1.5 Adagrad (Adaptive Gradient Descent) 8](#_Toc154221408)

[1.6 RMSprop (Root Mean Square) 9](#_Toc154221409)

[1.7 Adam 10](#_Toc154221410)

[1.8 Bảng so sánh các phương pháp 12](#_Toc154221411)

[CHƯƠNG 2: TÌM HIỂU VỀ CONTINUAL LEARNING VÀ TEST PRODUCTION KHI XÂY DỰNG MÔ HÌNH HỌC MÁY 14](#_Toc154221412)

[2.1 Continual Learning 14](#_Toc154221413)

[2.2 Test Production 15](#_Toc154221414)

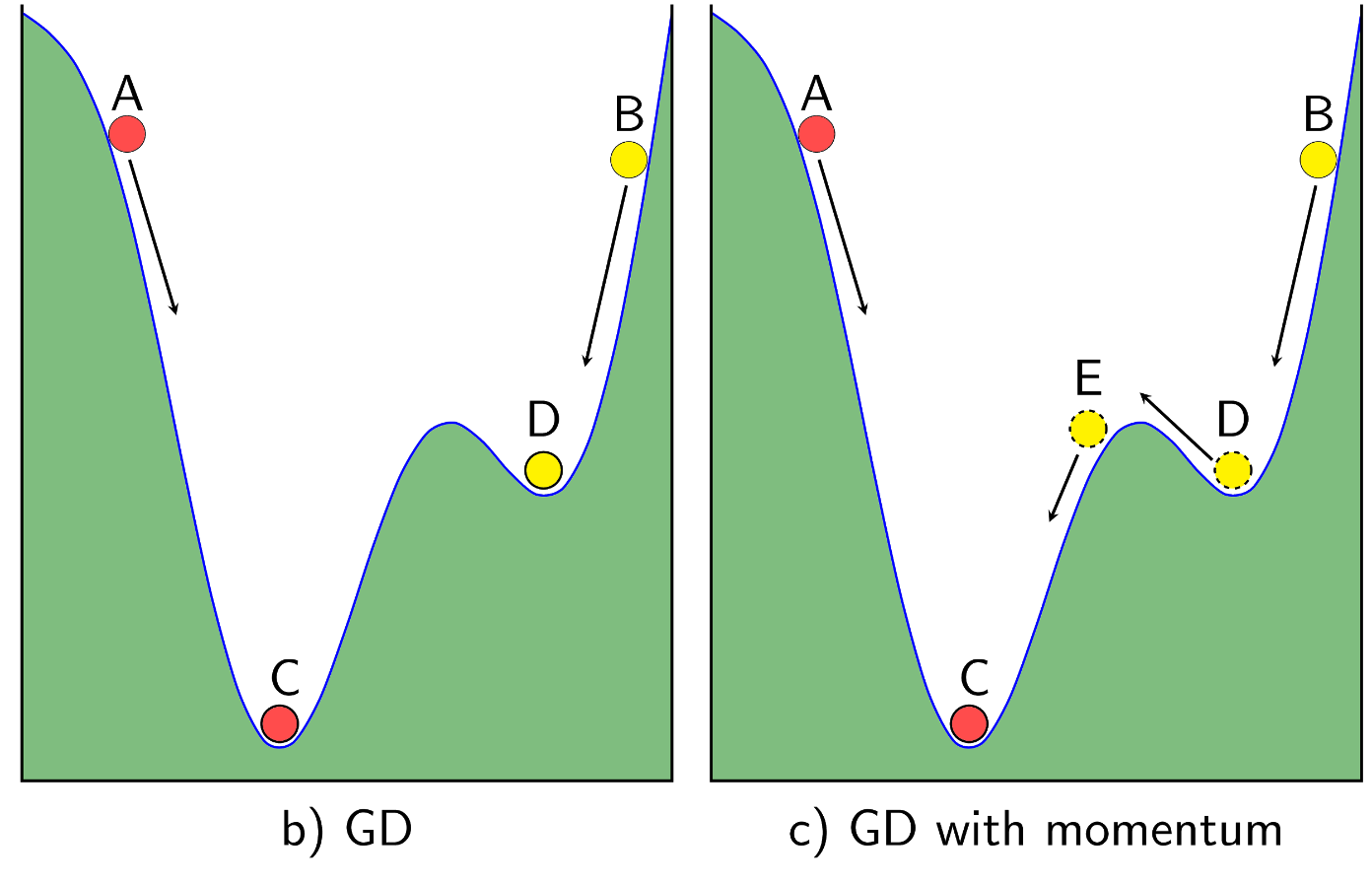
DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

**DANH MỤC HÌNH**

[Hình 1: Hình ảnh minh họa GD với momentum 6](#_Toc154221217)

CHƯƠNG 1 – TÌM HIỂU, SO SÁNH CÁC PHƯƠNG PHÁP OPTIMIZER TRONG HUẤN LUYỆN MÔ HÌNH HỌC MÁY

* 1. Mục tiêu của tối ưu hóa mô hình học máy
* Mục tiêu của tối ưu hóa mô hình học máy là tìm ra bộ trọng số cho mô hình sao cho hàm mất mát đạt giá trị nhỏ nhất hoặc tiệm cận giá trị tối thiểu. Mục tiêu này bao gồm:
  + Tối thiểu hóa hàm mất mát: Giảm thiểu giá trị của hàm mất mát.
  + Tránh Overfiting và Underfitting: Tránh việc mô hình quá khớp với dữ liệu đào tạo hoặc mô hình không đủ phức tạp để hiểu và dự đoán dữ liệu, đảm bảo mô hình có khả năng tổng quát hóa tốt trên dữ liệu mới mà không bị quá tương thích với dữ liệu huấn luyện.
  + Tăng tốc độ học: Điều chỉnh tốc độ học sao cho mô hình hội tụ một cách nhanh chóng và không gây ra dao động lớn.
  + Tối ưu hóa tính ổn định của mô hình: Đảm bảo rằng mô hình không quá nhạy cảm với các biến động nhỏ trong dữ liệu.
* Mục tiêu chính của tối ưu hóa là tìm ra trạng thái trọng số mô hình sao cho mô hình có thể thực hiện dự đoán chính xác trên dự liệu mới cà có khả năng quát hóa dữ liệu đã được đào tạo.
  1. Phương pháp Gradient Descent (GD)
* Gradient Descent (GD) là một thuật toán tối ưu hóa cơ bản được sử dụng rộng rãi trong machine learning, thuật toán này dùng để tìm điểm cực trị của hàm số, thuật toán này dựa trên ý tượng của quả bóng lăn xuống dốc để tìm điểm cực tiểu của hàm số.
* Cách thức hoạt động của thuật toán Gradien Descent như sau:
  + Khởi đầu với một điểm khởi tạo ngẫu nhiên cho các tham số của mô hình.
  + Tính đạo hàm riêng theo từng tham số và cập nhật các tham số theo hướng ngược lại với gradient của hàm mất mát.
  + Lặp lại quá trình này cho đến khi gradient gần bằng 0 hoặc đạt một giá trị cực nhỏ.
* Công thức chung của gradient descent: 
  + - Trong đó: η là tốc độ học (learning rate).
    - ∇θf(θ) là đạo hàm của hàm số tại một điểm θ bất kỳ.
* Ưu điểm
  + Thuật tóa này có thể áp dụng cho nhiều loại hàm mất mát khác nhau.
  + Thuật toán này có thể sử dụng các kỷ thuật biến đổi để giảm thiểu sai số và tăng hiệu suất. Ví dụ như mini-batch gradient descent.
  + Thuật toán này có thể kết hợp với các phương pháp khác để cải thiện kết quả, ví dụ như conjugate gradient method,…
  + Thuật toán này xử lý được vấn đề tối ưu hóa model neural network bằng cách cập nhật trọng số sau mỗi vòng lặp(epoch).
* Nhược điểm
  + Phụ thuộc vào nghiệm khởi tạo ban đầu và learning rate.
  + Có thể lạc vào điểm local minimum nếu không có phương pháp chọn điểm dừng phù hợp.
  + Thuật toán này có thể bị ảnh hưởng bởi sai số ngẫu nhiên trong quá trình gradient.
  1. Phương pháp Stochastic Gradient Descent (SGD)
* Định nghĩa: Đây là phiên bản của GD được áp dụng cho các mô hình có kích thước lớn hoặc không có gradient rõ ràng. Trong khi GD sẽ cập nhật θ 1 lần với mỗi lần duyệt một lượt qua tất cả các điểm trên toàn bộ dữ liệu, thì SGD cập nhật N lần trọng số θ với N là số lượng số điểm dữ liệu. Việc có N lần cập nhật θ sẽ làm tăng thời gian của epoch, tuy nhiên SGD sẽ chỉ cần một lượng epoch rất nhỏ để hoàn thành. Do đó mà SGD phù hợp với các bài có bài toán có tập dữ liệu lớn và phù hợp với online learning.
* Công thức cập nhật : 
  + - Trong đó: J(θ;xi;yi) là hàm mất mát với chỉ một cặp dữ liệu (input, label).
* Cách hoạt động:
  + Chọn một mini-batch: Thay vì sử dụng toàn bộ dữ liệu để tính gradient như trong Gradient Descent, SGD chỉ chọn ngẫu nhiên một số nhỏ các mẫu dữ liệu từ tập huấn luyện. Mini-batch này được sử dụng để tính gradient ước lượng.
  + Tính toán gradient ước lượng: Sử dụng mini-batch đã chọn, tính gradient của hàm mất mát đối với tham số hiện tại.
  + Cập nhật tham số: Sử dụng gradient ước lượng tính được, cập nhật tham số của mô hình theo hướng giảm độ dốc.
  + Lặp lại quá trình: quá trình này được lặp lại cho đến khi tiêu chí dừng được đáp ungws hoặc sau một số lần lặp cố định.
* Ưu điểm:
  + Là một phiên bản của gradient descent, phù hợp để xử lý tập dữ liệu có kích thước lớn.
  + Thời gian chạy nhanh: Do chỉ sử dụng mini-batch dữ liệu, SGD có thể hội tụ nhanh hơn so với Gradient Descent thông thường.
  + Linh hoạt: thích ứng nhanh chóng với các biến động trong dữ liệu.
  + Tránh overfitting: SGD sử dụng mini-batch và thay đổi dữ liệu đầu vào mỗi epoch giúp tránh overfitting.
* Nhược điểm:
  + Chưa giải quyết được 2 nhược điểm lớn của gradient descent ( điểm dữ liệu ban đầu và learning rate).
  1. Gradient Descent với Momentum
* Khái niệm: Momentum là một phương pháp tối ưu hóa được sử dụng việc hội tụ của thuật toán tối ưu hóa, đặc biệt là trong bài toán huấn luyện mô hình machine learning. Nó giải quyết được vấn đề của Gradient Descent thông thường, đặc biệt là khi có nhiều vùng phức tạp và điểm cực tiểu địa phương.



Hình 1: Hình ảnh minh họa GD với momentum

* Hình ảnh minh họa này là ý tưởng của thuật toán Momentum. Để tránh việc viên bi B dừng lại ở điểm D (local minimum) ta cần chuyền một vận tốc đủ lớn để viên bi có thể vượt qua D và đến được điểm E như trong ảnh.
* Công thức cập nhật tham số: 



* + - Trong đó: γ: parameter, thường có giá trị khoảng 0.9.

vt: vận tốc tại thời điểm trước đó.

∇θJ(θ): độ dốc của điểm trước đó.

* Cách hoạt động:
  + Bước 1: vt tích lũy thông tin từ gradient hiện tại và momentum trước đó. Nó giữ lại thông tin về hướng và tốc độ của các bước cập nhật trước.
  + Bước 2: θ được cập nhật sử dụng vt như một phần của nó. Vt giúp kiểm soát và giảm độ chập chờn trong quá trình tối ưu hóa.
* Ưu điểm:
  + Giải quyết được vấn đề không tiến được đến global minimum mà chỉ dừng lại ở local minimum.
  + Tăng tốc độ hội tụ: Gradient Descent với momentum thường hội tụ nhanh hơn so với GD thông thường, đặc biệt trên các bề mặt hàm mất mát có nhiều độ chập chờn.
* Nhược điểm:
  + Cần phải chọn tham số tối ưu cho hệ số momentum.
  + Có thể bị kẹt ở điểm cực địa phương (local minimum) nếu chọn hệ số quá lớn.
  1. Adagrad (Adaptive Gradient Descent)
* Khái niệm: Thuật toán Adagrad là phương pháp tối ưu hóa được sử dụng trong máy học đẻ cập nhật trọng số của mô hình theo cách có thể điều chỉnh learning rate tùy thuộc vào từng tham số. Adagrad được tạo ra để đối phó với vấn đề của learning rate cố định trong các thuật toán tối ưu hóa khác.
* Cách hoạt động:
  + Tính gradient: Tính gradient của hàm mất mát đối với từng tham số *θ* tại thời điểm hiện tại *t*. Điều này thường được thực hiện bằng cách sử dụng một số dữ liệu huấn luyện (mini-batch) ngẫu nhiên.
  + Tính toán trung bình các bình phương Gradient:
    - Tính tổng bình phương các gradient đã tính được cho mỗi tham số từ thời điểm bắt đầu đến thời điểm hiện tại. Điều này được thực hiện để tính toán mức độ biến động của gradient. 
    - Ở đây, *Gt*,*i*​ là tổng bình phương các gradient cho tham số thứ *i* tại thời điểm *t*, ∇*J*(*θt*,*i*​) là gradient tương ứng.
  + Cập nhật tham số:
    - Cập nhật từng tham số của mô hình bằng cách sử dụng learning rate được điều chỉnh theo bình phương của gradient. 
    - Ở đây, *η* là learning rate ban đầu, *Gt*,*i*​ là tổng bình phương gradient, và *ϵ* là một số nhỏ được thêm vào để tránh chia cho 0.
* Ưu điểm:
  + Ưu điểm rõ ràng của Adagrad là tránh việc điều chỉnh learning rate bằng tay, chỉ cần để tốc độ học default là 0.01 thì thuật toán sẽ tự điều chỉnh.
  + Hiệu quả trong việc xử lý các tham số có độ lớn khác nhau.
* Nhược điểm:
  + Có thể bị kẹt ở điểm cực tiểu địa phương (local minimum)
  + Learning rate giảm dần theo thời gian, dẫn đến việc hội tụ chậm hơn hoặc không hội tụ nếu tốc độ học tập quá nhỏ.
  + Yêu cầu bộ nhớ lớn: Do cần lưu trữ lịch sử của gradient cho từng tham số, Adagrad có thể yêu cầu một lượng lớn bộ nhớ.
  1. RMSprop (Root Mean Square)
* Khái niệm: Thuật toán RMSprop là phương pháp tối ưu hóa được tạo ra để cải thiện khả năng hội tụ của mô hình trong quá trình huấn luyện. Thuật toán này giữ lại các ưu điểm của Adagrad vừa nêu trên trong việc tự điều chỉnh learning rate, đồng thời giải quyết vấn đề tỷ lệ học giảm dần của Adagrad bằng cách chia tỷ lệ học cho trung bình của bình phương gradient.
* Cách hoạt động:
  + Tính gradient: Tính gradient của hàm mất mát đối với từng θ tại thời điểm t.
  + Tính toán trung bình các bình phương gradient: Tính EMA( exponentially weighted moving average) của bình phương các gradient đã tính được cho mỗi tham số từ thời điểm bắt đầu đến hiện tại.



* + - Trong đó: *E*[*g*2]*t*​ là EMA của bình phương gradient cho tham số *θ*, *β* là hệ số giảm trọng số (thường là khoảng 0.9), và ∇*J*(*θt*​) là gradient tương ứng.
  + Cập nhật tham số: cập nhật tham số của mô hình bằng cách sử dụng learning rate điều chỉnh theo căn bậc hai của EMA của bình phương gradient.



* + - Trong đó: *ϵ* là một số nhỏ được thêm vào để tránh chia cho 0.
* Ưu điểm:
  + Giải quyết được vấn đề learning rate giảm dần của Adagrad.
  + Phù hợp với dữ liệu thưa vì nó không tích lũy quá nhiều thông tin về gradient.
* Nhược điểm:
  + Có khả năng bị Overshoot.
  + Trong một số trường hợp, RMSprop vẫn có thể gặp vấn đề với khả năng suy giảm learning rate quá nhanh.
  1. Adam
* Khái niệm: Adam Optimizer ( Adaptive Moment Estimation) là một thuật toán tối ưu được sử dụng phổ biến. Thuật toán này kết hợp cả Momentum và RMSprop, tận dụng ưu điểm của cả hai để tối ưu hóa hàm mất mát. Tương tự như RMSProp, Adam xem xét khoảng cách thứ hai của độ dốc, nhưng không giống như RMSProp, nó tính toán phương sai không tập trung vào
* Cách hoạt động:
  + Momentum: Momentum giúp giảm hiện tượng dap động và tăng tốc độ hội tụ bằng cách tính toán trung bình có trọng số của gradient trước đó.



Trong đó: mt là momentum tại thời gian t.

*β*1​ là hệ số giảm momentum( thường khoảng 0.9)

*gt*​ là gradient tại bước thời gian t.

* + RMSprop: giảm độ lớn của gradient bằng cách chia nó cho một độ lớn tích lũy trước đó. Bước cập nhật của RMSprop như sau:



Trong đó: vt là độ lớn tích lũy tại bước thời gian t.

*β*2​ là hệ số giảm dần độ lớn tích lũy (thường là 0.999)

* + Cập nhật tham số sau khi tính momentum và RMSprop:



Trong đó: *θt*​ là tham số tại bước thời gian t.

*η* là learning rate

*ϵ* là một giá trị nhỏ để tránh chia cho 0.

* Ưu điểm:
  + Tối ưu hóa nhanh và hiệu quả hơn so với các thuật toán khác.
  + Có thể sử dụng cho các mô hình lớn với số lượng tham số lớn.
  + Xử lý dữ liệu và độ dốc nhiễu.
  + Ít bộ nhớ.
* Nhược điểm:
  + Có thể dẫn đến overfitting nếu sử dụng trên dữ liệu có kích thước quá nhỏ.
  + Khả năng tăng tốc chậm: Trong vài trường hợp, Adam có thể tăng tốc chậm hơn các phương pháp tối ưu hóa khác trong những giai đoạn đầu quá trình học.
  1. Bảng so sánh các phương pháp

|  |  |  |  |
| --- | --- | --- | --- |
| Phương pháp | Tính chất chính | Ứng dụng | Đặc điểm |
| Gradient Descent (GD) | * Cơ bản, dùng toàn bộ dữ liệu * Dễ hiểu | * Hồi quy tuyến tính, mô hình đơn giản. * Không hợp với dữ liệu lớn. | * Tốt cho dữ liệu nhỏ, không nhạy cảm với dao động. * Cần nhiều tài nguyên nếu tập dữ liệu lớn. |
| Stochastic Gradient Descent (SGD) | * Sử dụng mẫu ngẫu nhiên. * Thích hợp cho tập dữ liệu lớn. * Dễ bị nhiễu. | * Phân loại, deep learning. * Huấn luyện mô hình nhanh. | * Nhanh, phù hợp với dữ liệu lớn, dao động cao. * Độ dao động cao, có thể mắc kẹt ở local minimum. |
| GD với Momentum | * Kết hợp momentum cho độ ổn định * Tăng tốc học. | * Hội tụ nhanh, phù hợp nhiều bài toán. | * Giảm độ dao động, tăng tốc quá trình học. * Cần điều chỉnh thêm siêu tham số, như hệ số momentum. |
| Adagrad | * Learning rate thích ứng với từng tham số. * Phù hợp với dự liệu có biến động lớn. | * Bài toán với dữ liệu rãi rác. * Huấn luyện mô hình trên dữ liệu thưa. | * Hiệu quả với dữ liệu biến động. * Tích lũy bình phương gradient, giảm learning rate. |
| RMSprop | * Giảm độ giao động của learning rate. * Hiệu quả với dữ liệu rải rác. | * Dữ liệu thưa, deep learning. * Phù hợp cho các tác vụ đa nhiệm | * Thích hợp cho dữ liệu biến động. * Điều chỉnh learning rate theo từng tham số. |
| Adam | * Kết hợp ưu điểm của Momentum và RMSprop * Phù hợp với nhiều bài toán và loại dữ liệu. * Hiệu suất tốt với mô hình học sâu. | * Các bài toán yêu cầu tối ưu hóa linh hoạt và hiệu quả | * Hiệu quả trên nhiều loại mô hình và dữ liệu. * Có thể bị overfitting trên tập dữ liệu quá nhỏ |

CHƯƠNG 2: TÌM HIỂU VỀ CONTINUAL LEARNING VÀ TEST PRODUCTION KHI XÂY DỰNG MÔ HÌNH HỌC MÁY

2.1 Continual Learning

* Khái niệm:
  + Continual learning trong machine learning là một quá trình mà mô hình học dữ liệu mới mà không cần phải đào tạo lại. Mục tiêu của continual learning là làm cho mô hình có thể thích ứng với môi trường thay đổi và nguồn dữ liệu mới mà không gặp vấn đề quên thông tin quan trong đã học.
* Quá trình continual learning:
  + Khởi tạo: khởi tạo một mô hình học máy ban đầu.
  + Đào tạo ban đầu và thu thập dữ liệu: Huấn luyện mô hình trên tập dữ liệu ban đầu. Thu thập và lưu trữ dữ liệu liên quan về trạng thái và biểu diễn nội tại của mô hình.
  + Tích hợp dữ liệu mới: Khi có dữ liệu mới thì tích hợp nó vào quá trình đào tạo của mô hình mà không ảnh hưởng nhiều đến dữ liệu đã học.
  + Phòng tránh Catastrophic Forgetting: Sự dụng các phương pháp như Elastic Weight Consolidation (EWC) hoặc Synaptic Intelligence (SI) để bảo toàn thông tin quan trọng trong quá khứ và giảm rủi ro quên thông tin.
  + Rehearsal: Lưu trữ các mẫu dữ liệu trong quá khứ và đào tạo lại mô hình để duy trì thông tin cũ.
  + Regularization Techniques: Áp dụng kỹ thuật chính quy để kiểm soát và giảm thiểu thay đổi không mong mốn của trọng số.
  + Dynamic Architectures: Cập nhật kiến trúc của mô hình động khi cần thiết để chứa dữ liệu mới mà không làm giảm hiệu suất.
  + Meta – learning: Tích hợp các phương pháp meta-learning để mô hình nhanh chóng thích ứng với dữ liệu mới.
  + Kiểm tra và đánh giá: Tiến hành kiểm tra định kỳ trên cả dữ liệu cũ và mới để đảm bảo rằng mô hình duy trì hiệu suất ở cả hai.
* Ưu điểm:
  + Tích hợp kiến thức liên tục.
  + Giúp mô hình có hiệu suất mạnh mẽ và chính xác hơn.
  + Lưu trữ thông tin.
  + Khả năng thích ứng.
* Nhược điểm:
  + Độ phức tạp cao.
  + Khả năng bị overfitting: Nếu không quản lý chặt chẽ, mô hình có thể dễ dàng quá mức dựa vào dữ liệu mới và có nguy cơ bị overfitting.
  + Yêu cầu lượng dữ liệu lớn.

2.2 Test Production

* Khái niệm cơ bản:
  + Test production tròn machine learning là quá trình kiểm tra và đánh giá một mô hình học máy trước khi triển khai mô hình vào thế giới thực. Việc này bao gồm các quá trình kiểm tra như kiểm tra hiệu suất, độ chính xác, khả năng thích ứng với của mô hình với tập dữ liệu mới và được biết đến. Mục đích của quá trình này là đảm bảo rằng các mô hình đã được đào tạo hoat động hiệu quả trên và tổng quát khi triển khai trong các ứng dụng thực tế.
* Công cụ và kỹ thuật được sử dụng cho test Production:
  + Mlflow: MLflow là một nền tảng quản lý toàn bộ quá trình phát triển mô hình machine learning (ML). Nó giúp bạn theo dõi và quản lý các thí nghiệm, đăng ký mô hình, đóng gói mô hình, và theo dõi sự thay đổi phiên bản và hiệu suất của mô hình.
  + Neptune.ai: Neptune.ai là nền tảng giúp tổ chức thí nghiệm machine learning một cách hiệu quả. Nó giúp bạn tổ chức thí nghiệm thành sổ tay hoặc pipeline, và thực hiện theo dõi kết quả với bảng điều khiển hoặc báo cáo
  + mlqa: mlqa là một khung kiểm thử mô hình machine learning bằng cách sử dụng truy vấn ngôn ngữ tự nhiên (NLU). Bạn có thể viết trường hợp kiểm thử bằng ngôn ngữ tự nhiên tiếng Anh.
  + Mlcommons: là một thư viện cung cấp các công cụ hữu ích cho nhiều tác vụ trong lĩnh vực machine learning. Nó giúp xử lý dữ liệu, đặc trưng hóa, và đánh giá mô hình.
* Lợi ích của Test Production:
  + Cải thiện chất lượng và độ chính xác của mô hình.
  + Giảm thiểu rủi ro: Các vấn đề về ỗi có thể được phát hiện và giải quyết ngay từ giai đoạn triển khai sớm.
  + Nâng cao hiệu quả và năng suất.
* Nhược điểm:
  + Yều cầu tài nguyên và thời gian: Việc triển khai tets Production có thể kéo dài thời gian vì quá trình kiểm thử và điều chỉnh liên tục.
  + Việc triển khai test productiion có thể tăng chi phí phát triển.

**TÀI LIỆU THAM KHẢO**

**Tiếng Việt**

1. https://machinelearningcoban.com/2017/01/12/gradientdescent/

**Tiếng Anh**

1. https://fullstackdeeplearning.com/course/2022/lecture-6-continual-learning/
2. https://en.wikipedia.org/wiki/Machine\_learning