TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**Nhập Môn Học Máy**

**BÁO CÁO CUỐI KỲ**

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

*Người thực hiện*: **Trang Sĩ Nam – 521H0276**

Lớp **: 21H50302**

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**



**Nhập Môn Học Máy**

**BÁO CÁO CUỐI KỲ**

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

*Người thực hiện*: **TRANG SĨ NAM – 521H0276**

Lớp **: 21H50302**

Khoá  **: 25**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

LỜI CẢM ƠN

Em xin chân thành cảm ơn thầy Lê Anh Cường - người trực tiếp giảng dạy những kiến thức cơ bản và cần thiết về Nhập môn Học Máy. Nhờ sự hướng dẫn, giúp đỡ và điều kiện thuận lợi, em đã hoàn thành được báo cáo cuối kì.

Em chân thành cảm ơn!

**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 PHS 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 23 tháng 12 năm 2023*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Trang Sĩ Nam*

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 nghiên cứu:

* So sánh các phương pháp Optimizer trong huấn luyện mô hình.
* Tìm hiểu về Continual Learning và Test Production khi xây dựng một giải pháp học máy

MỤC LỤC

[LỜI CẢM ƠN i](#_Toc154259283)

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN iii](#_Toc154259284)

[TÓM TẮT iv](#_Toc154259285)

[MỤC LỤC 1](#_Toc154259286)

[CHƯƠNG 1 – OPTIMIZER 2](#_Toc154259287)

[1.1 Optimizer 2](#_Toc154259288)

[1.1.1 Khái quát Optimizer 2](#_Toc154259289)

[1.1.2 Các phương pháp Optimizer 2](#_Toc154259290)

[1.1.3 So sánh các phương pháp Optimizer 4](#_Toc154259291)

[CHƯƠNG 2 – CONTINUAL LEARNING VÀ TEST PRODUCTION 6](#_Toc154259292)

[2.1 Continual Learning 6](#_Toc154259293)

[2.2 Test Production 7](#_Toc154259294)

[TÀI LIỆU THAM KHẢO 9](#_Toc154259295)

CHƯƠNG 1 – OPTIMIZER

1.1 Optimizer

1.1.1 Khái quát Optimizer

Trong học máy, quá trình huấn luyện mô hình thường được thực hiện bằng cách sử dụng một thuật toán tối ưu hóa để tìm các tham số mô hình tối ưu hóa hàm mục tiêu. Hàm mục tiêu thường là một hàm mất mát, đo lường mức độ sai lệch giữa dự đoán của mô hình và giá trị thực của dữ liệu. Các thuật toán tối ưu hóa được sử dụng trong học máy được gọi là optimizer.

Có nhiều loại optimizer khác nhau, mỗi loại có ưu nhược điểm riêng. Một số optimizer phổ biến bao gồm:

* Stochastic Gradient descent: Đây là một trong những optimizer đơn giản và hiệu quả nhất. Gradient descent sử dụng đạo hàm của hàm mục tiêu để tính ra hướng đi cần thiết để giảm thiểu hàm mục tiêu.
* Momentum: Momentum là một biến thể của gradient descent, sử dụng thêm một số thông tin từ các bước trước để cải thiện tốc độ hội tụ của optimizer.
* Adagrad: Adagrad là một optimizer adaptiv, điều chỉnh tốc độ học tập của các tham số dựa trên mức độ thay đổi của hàm mục tiêu.
* RMSprop: RMSprop là một optimizer adaptiv khác, tương tự như Adagrad nhưng sử dụng bình phương của đạo hàm thay vì đạo hàm.
* Adam: Adam là một optimizer kết hợp các ưu điểm của gradient descent, momentum và RMSprop.

Cách lựa chọn optimizer phụ thuộc vào nhiều yếu tố, bao gồm loại mô hình, kích thước dữ liệu, hàm mục tiêu và các ràng buộc về thời gian và tài nguyên tính toán.

1.1.2 Các phương pháp Optimizer

Gradient Descent

* Gradient descent là một trong những optimizer đơn giản và hiệu quả nhất. Gradient descent sử dụng đạo hàm của hàm mục tiêu để tính ra hướng đi cần thiết để giảm thiểu hàm mục tiêu.
* Định nghĩa
* Giả sử hàm mục tiêu là f(x), và x là vector tham số của mô hình. Gradient descent sẽ cập nhật tham số x theo công thức sau:

x = x - t \* grad(f(x))

Trong đó,

t là tốc độ học tập,

grad(f(x)) là đạo hàm của hàm mục tiêu tại điểm x.

Momentum

* Momentum là một biến thể của gradient descent, sử dụng thêm một số thông tin từ các bước trước để cải thiện tốc độ hội tụ của optimizer.
* Định nghĩa
* Momentum sẽ cập nhật tham số x theo công thức sau:

x = x + v \* t

* Trong đó, v là vector động lượng, được tính theo công thức sau:

v = v - t \* grad(f(x))

Adagrad

* Adagrad là một optimizer adaptiv, điều chỉnh tốc độ học tập của các tham số dựa trên mức độ thay đổi của hàm mục tiêu.
* Định nghĩa
* Adagrad sẽ cập nhật tham số x theo công thức sau:

x = x - t \* grad(f(x)) / (sqrt(g) + eps)

Trong đó,

g là vector lưu trữ tổng bình phương của đạo hàm của hàm mục tiêu tại tất cả các bước huấn luyện.

RMSprop

* RMSprop là một optimizer adaptiv khác, tương tự như Adagrad nhưng sử dụng bình phương của đạo hàm thay vì đạo hàm.
* Định nghĩa
* RMSprop sẽ cập nhật tham số x theo công thức sau:

x = x - t \* grad(f(x)) / (sqrt(g\_t) + eps)

Trong đó,

g\_t là vector lưu trữ bình phương của đạo hàm của hàm mục tiêu tại bước huấn luyện t.

Adam

* Adam là một optimizer kết hợp các ưu điểm của gradient descent, momentum và RMSprop.
* Định nghĩa
* Adam sẽ cập nhật tham số x theo công thức sau:

x = x - t \* (m\_t / (sqrt(v\_t) + eps))

Trong đó,

m\_t và v\_t là vector lưu trữ tổng và bình phương của đạo hàm của hàm mục tiêu tại tất cả các bước huấn luyện.

1.1.3 So sánh các phương pháp Optimizer

Gradient Descent

Ưu điểm

* Gradient descent là một phương pháp đơn giản và dễ hiểu.
* Gradient descent có thể được sử dụng để huấn luyện các mô hình học máy có kích thước lớn.

Nhược điểm

* Gradient descent có thể có tốc độ hội tụ chậm, đặc biệt là đối với các hàm mục tiêu có nhiều cục bộ tối ưu.

Momentum

Ưu điểm

* Momentum có thể cải thiện tốc độ hội tụ của gradient descent.
* Momentum có thể giúp mô hình tránh được các cục bộ tối ưu.

Nhược điểm

* Momentum có thể khó điều chỉnh hơn gradient descent.

Adagrad

Ưu điểm

* Adagrad có thể thích ứng với các tham số của mô hình.
* Adagrad có thể giúp mô hình tránh được các cục bộ tối ưu.

Nhược điểm

* Adagrad có thể dẫn đến overfitting.

RMSprop

Ưu điểm

* RMSprop tương tự như Adagrad, nhưng có thể ổn định hơn.
* RMSprop có thể giúp mô hình tránh được các cục bộ tối ưu.

Nhược điểm

* RMSprop có thể dẫn đến overfitting.

Adam

Ưu điểm

* Adam có thể thích ứng với các tham số của mô hình.
* Adam có thể giúp mô hình tránh được các cục bộ tối ưu.
* Adam có thể có tốc độ hội tụ nhanh hơn các optimizer khác.

Nhược điểm

* Adam có thể khó điều chỉnh hơn các optimizer khác.

CHƯƠNG 2 – CONTINUAL LEARNING VÀ TEST PRODUCTION

2.1 Continual Learning

Continual Learning (hay còn được gọi là Lifelong Learning hoặc Incremental Learning) là một lĩnh vực trong Machine Learning (học máy) mà mô hình học được liên tục cập nhật và mở rộng kiến thức của mình khi đối mặt với dữ liệu mới mà không quên đi kiến thức đã học trước đó. Trong bối cảnh này, một giải pháp học máy được xây dựng để giải quyết một bài toán nào đó sẽ cần có khả năng tiếp tục học và sử dụng kiến thức đã học được từ các tác vụ trước đó để hỗ trợ việc học và tăng cường hiệu suất của mô hình.

Việc áp dụng Continual Learning trong xây dựng giải pháp học máy cho một bài toán cụ thể có thể được thực hiện theo các bước sau:

1. Thiết kế mô hình: Xác định kiến trúc mô hình học máy phù hợp cho bài toán cần giải quyết. Kiến trúc mô hình nên có khả năng mở rộng và tăng cường để chứa thông tin mới và hạn chế việc quên đi kiến thức cũ.
2. Xác định phương pháp học liên tục: Chọn phương pháp học liên tục phù hợp để đảm bảo mô hình có khả năng học từ dữ liệu mới mà không ảnh hưởng đến hiệu suất của các tác vụ đã học trước đó. Một số phương pháp học liên tục phổ biến bao gồm Elastic Weight Consolidation (EWC), Online EWC, Synaptic Intelligence (SI), và Generative Replay.
3. Quản lý bộ nhớ: Để giải quyết vấn đề quên đi kiến thức cũ, cần có cơ chế để lưu trữ và quản lý các thông tin quan trọng từ các tác vụ trước đó. Các phương pháp như Experience Replay và Memory Regularization có thể được sử dụng để lưu trữ và truy xuất lại dữ liệu cũ.
4. Đánh giá và điều chỉnh: Thực hiện việc đánh giá và điều chỉnh mô hình sau khi học từ dữ liệu mới. Điều này có thể bao gồm việc đo lường hiệu suất của mô hình trên các tác vụ đã học trước đó và điều chỉnh các tham số để tối ưu hóa hiệu suất tổng thể.
5. Lặp lại quá trình: Tiếp tục lặp lại các bước trên khi có dữ liệu mới hoặc khi có nhu cầu mở rộng khả năng học của mô hình.

Qua việc áp dụng Continual Learning trong xây dựng giải pháp học máy, mô hình sẽ có khả năng học tập liên tục và tích lũy kiến thức từ các tác vụ trước đó, đồng thời giải quyết hiệu quả các bài toán mới mà không cần phải huấn luyện lại từ đầu. Điều này đặc biệt hữu ích trong các hệ thống học máy thời gian thực hoặc trong các môi trường có dữ liệu thay đổi liên tục.

2.2 Test Production

Test Production (sản xuất kiểm thử) là quá trình tạo ra các bộ dữ liệu kiểm thử để đánh giá hiệu suất của một giải pháp học máy trong việc giải quyết một bài toán cụ thể. Khi xây dựng một giải pháp học máy, quá trình test production rất quan trọng để đảm bảo tính đáng tin cậy và hiệu quả của mô hình. Dưới đây là một số khía cạnh cần xem xét khi thực hiện test production:

1. Lựa chọn bộ dữ liệu kiểm thử: Đầu tiên, cần chọn một bộ dữ liệu kiểm thử phù hợp với bài toán được giải quyết. Bộ dữ liệu này nên phản ánh các trường hợp đại diện và đa dạng của bài toán, bao gồm các tình huống biên, dữ liệu nhiễu, và các yếu tố khác có thể gặp trong thực tế.
2. Chuẩn bị dữ liệu: Dữ liệu kiểm thử cần được chuẩn bị một cách cẩn thận. Điều này bao gồm việc tiền xử lý, xử lý nhiễu, chuẩn hóa dữ liệu và phân chia dữ liệu thành tập huấn luyện và tập kiểm thử. Bước chuẩn bị dữ liệu này đảm bảo rằng dữ liệu kiểm thử có tính thống nhất và phù hợp với mô hình học máy.
3. Định nghĩa các phép đo hiệu suất: Trước khi tiến hành kiểm thử, cần xác định các phép đo hiệu suất cụ thể để đánh giá mô hình. Điều này có thể bao gồm độ chính xác, độ phân loại, độ đo F1, mất mát, hay các phép đánh giá khác tùy thuộc vào bài toán.
4. Thực hiện kiểm thử: Sau khi chuẩn bị dữ liệu và định nghĩa các phép đo hiệu suất, tiến hành thực hiện kiểm thử bằng cách đưa dữ liệu kiểm thử vào mô hình học máy và đo lường hiệu suất. Đảm bảo rằng quy trình kiểm thử được thực hiện theo cách chính xác và có kế hoạch để đảm bảo sự nhất quán và khách quan.
5. Đánh giá kết quả kiểm thử: Cuối cùng, đánh giá kết quả kiểm thử và xem xét hiệu suất của mô hình. Phân tích các phép đo hiệu suất, so sánh với các mô hình khác hoặc các tiêu chuẩn thực tế, và suy luận về độ tin cậy và hiệu quả của giải pháp học máy.

Quá trình test production là một phần quan trọng trong việc xây dựng giải pháp học máy. Nó giúp đảm bảo rằng mô hình đã được phát triển đáng tin cậy, có hiệu suất tốt và phù hợp với yêu cầu của bài toán cụ thể.

# TÀI LIỆU THAM KHẢO

**Tiếng Việt**

|  |  |
| --- | --- |
| [1] | Trần Trung Trực, “Optimizer- Hiểu sâu về các thuật toán tối ưu ( GD,SGD,Adam,..)”, 17/10/2020. Available: [Optimizer- Hiểu sâu về các thuật toán tối ưu ( GD,SGD,Adam,..) (viblo.asia)](https://viblo.asia/p/optimizer-hieu-sau-ve-cac-thuat-toan-toi-uu-gdsgdadam-Qbq5QQ9E5D8) |
| [2] | Thi Nguyễn Thi, "Tại sao bạn nên kiểm thử trên môi trường Production?," 20/3/2023. Available [Tại sao bạn nên kiểm thử trên môi trường Production? (viblo.asia)](https://viblo.asia/p/tai-sao-ban-nen-kiem-thu-tren-moi-truong-production-WAyK8oOm5xX). |

**Tiếng Anh**

|  |  |
| --- | --- |
|  |  |
| [3] | GitBook, " Introduction to Continual Learning", 06/03/2021. Available: [Introduction to Continual Learning - Wiki (continualai.org)](https://wiki.continualai.org/the-continualai-wiki/introduction-to-continual-learning) |
| [4] | Goku Mohandas, " Testing Machine Learning Systems: Code, Data and Models" .Available: [Testing Machine Learning Systems: Code, Data and Models - Made With ML](https://madewithml.com/courses/mlops/testing/) |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |