TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**TRẦN THỊ NGỌC ÁNH - 52000008**

**BÁO CÁO CUỐI KỲ**

**NHẬP MÔN  
HỌC MÁY**

**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**



**TRẦN THỊ NGỌC ÁNH - 52000008**

**BÁO CÁO CUỐI KỲ**

**NHẬP MÔN  
HỌC MÁY**

Người hướng dẫn

**PGS TS. Lê Anh Cường**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

**LỜI CẢM ƠN**

Em xin chân thành cảm ơn Thầy Lê Anh Cường và Trường Đại học Tôn Đức Thắng đã hướng dẫn và hỗ trợ em trong môn học Nhập môn Học máy. Những kiến thức tiếp thu được từ Thầy không chỉ là tiền đề để em hoàn thành báo cáo này mà còn là hành trang cho bản thân em trong tương lai.

Vì kiến thức là vô hạn nên báo cáo không thể tránh khỏi những sai sót, em rất mong nhận được những phản hồi của Thầy để em hoàn thiện kiến thức cho bản thân hơn.

Em xin chân thành cảm ơn!

*TP. Hồ Chí Minh, ngày 20 tháng 12 năm 2023*

*Tác giả*

*(Ký tên và ghi rõ họ tên)*

*Trần Thị Ngọc Ánh*

**CÔNG TRÌNH 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 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 Dự 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 Dự 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 20 tháng 12 năm 2023*

*Tác giả*

*(Ký tên và ghi rõ họ tên)*

*Trần Thị Ngọc Ánh*

**TÓM TẮT**

Trình bày một bài nghiên cứu, đánh giá về các vấn đề sau:

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

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**

[DANH MỤC BẢNG BIỂU v](#_Toc2433)

[CHƯƠNG 1. CÁC PHƯƠNG PHÁP OPTIMIZER TRONG HUẤN LUYỆN MÔ HÌNH HỌC MÁY 1](#_Toc9754)

[1.1 Khái niệm về optimizer 1](#_Toc21637)

[1.2 So sánh các phương pháp optimizer 1](#_Toc26703)

[CHƯƠNG 2. CONTINUAL LEARNING VÀ TEST PRODUCTION 6](#_Toc3266)

[2.1 Continual Learning 6](#_Toc2343)

[2.2 Test Production 7](#_Toc31534)

[2.3 Ứng dụng của Continual Learning và Test Production 8](#_Toc16937)

[TÀI LIỆU THAM KHẢO 10](#_Toc24051)

# DANH MỤC BẢNG BIỂU

[Bảng 1 .1 So sánh các phương pháp optimizer 5](#_Toc12358)

# CÁC PHƯƠNG PHÁP OPTIMIZER TRONG HUẤN LUYỆN MÔ HÌNH HỌC MÁY

## Khái niệm về optimizer

Trong lĩnh vực học máy, optimizer là một thuật toán được sử dụng để cập nhật các tham số của một mô hình học máy trong quá trình huấn luyện. Các tham số này là các giá trị ảnh hưởng đến cách mô hình hoạt động. Việc cập nhật các tham số này giúp mô hình học cách thực hiện nhiệm vụ được giao một cách tốt hơn.

Các phương pháp tối ưu hóa đóng vai trò quan trọng trong việc tìm ra các tham số tối ưu của mô hình. Các phương pháp tối ưu hóa sẽ sử dụng một hàm mục tiêu để đánh giá mô hình và sau đó điều chỉnh các tham số của mô hình để giảm thiểu hàm mục tiêu đó.

## So sánh các phương pháp optimizer

Có nhiều phương pháp tối ưu hóa khác nhau, mỗi phương pháp đều có những ưu điểm và nhược điểm riêng. Dưới đây là một số phương pháp tối ưu hóa phổ biến trong huấn luyện mô hình học máy:

|  |  |  |  |
| --- | --- | --- | --- |
| Phương pháp | Mô tả | Ưu điểm | Nhược điểm |
| Gradient Descent (GD) | Sử dụng đạo hàm của hàm mục tiêu để tính toán hướng đi cần thiết nhằm giảm thiểu hàm mục tiêu.  Ý tưởng của gradient descent là bắt đầu từ một điểm ngẫu nhiên trong không gian tham số của mô hình. Sau đó, phương pháp này sẽ sử dụng đạo hàm của hàm mục tiêu tại điểm hiện tại để tính toán hướng đi cần thiết để giảm thiểu hàm mục tiêu. Phương pháp này sẽ tiếp tục di chuyển theo hướng đó cho đến khi đạt được điểm tối ưu của hàm mục tiêu. | - Đơn giản và dễ hiểu nhất.  - Không yêu cầu nhiều tài nguyên tính toán. | - Học chậm.  - Dễ bị kẹt trong các điểm tối ưu cục bộ. |
| Stochastic Gradient Descent (SGD) | Sử dụng rộng rãi trong huấn luyện mô hình học máy, đặc biệt là trong các mạng nơ-ron sâu.  Giống như Gradient Descent (lướt dốc), SGD tìm kiếm giá trị tối thiểu của một hàm mục tiêu (thường là lỗi của mô hình) bằng cách di chuyển các tham số của mô hình theo hướng giảm dần của hàm đó. Tuy nhiên, thay vì tính toán đạo hàm của hàm mục tiêu cho toàn bộ tập dữ liệu huấn luyện, SGD chỉ sử dụng một mẫu dữ liệu duy nhất mỗi lần cập nhật tham số. | - Học nhanh hơn GD.  - Thích ứng tốt với dữ liệu lớn.  - Giảm khả năng overfitting. | - Bất ổn định.  - Dễ bị kẹt trong điểm tối ưu cục bộ. |
| Mini-Batch Gradient Descent (mini-BGD) | Sử dụng một phần dữ liệu huấn luyện để tính toán đạo hàm của hàm mục tiêu.  Kích thước của mini-batch thường được chọn là một số nguyên tố nhỏ, chẳng hạn như 32, 64, hoặc 128. Kích thước mini-batch càng nhỏ thì mô hình sẽ học hỏi nhanh hơn, nhưng cũng có khả năng bị quá khớp (overfitting) cao hơn. Kích thước mini-batch càng lớn thì mô hình sẽ học hỏi chậm hơn, nhưng cũng có khả năng bị quá khớp thấp hơn. | - Học nhanh hơn SGD.  - Giảm khả năng overfitting.  - Có thể áp dụng cho các mô hình học máy lớn. | - Có thể bị kẹt trong các điểm tối ưu cục bộ.  - Yêu cầu nhiều tài nguyên tính toán hơn SGD. |
| Momentum | Sử dụng trọng lượng của hướng đi trước đó để tính toán hướng đi hiện tại.  Ý tưởng của momentum là bắt đầu từ một điểm ngẫu nhiên trong không gian tham số của mô hình. Sau đó, phương pháp này sẽ sử dụng đạo hàm của hàm mục tiêu tại điểm hiện tại để tính toán hướng đi cần thiết để giảm thiểu hàm mục tiêu. Tuy nhiên, thay vì cập nhật các tham số của mô hình theo hướng đi này ngay lập tức, phương pháp momentum sẽ sử dụng một trọng lượng để pha trộn hướng đi này với hướng đi trước đó. | - Giúp mô hình vượt qua các điểm tối ưu cục bộ.  - Học nhanh hơn SGD.  - Ít bị ảnh hưởng bởi nhiễu trong dữ liệu. | - Có thể dẫn đến dao động trong quá trình tối ưu hóa.  - Khó điều chỉnh các tham số. |
| Adagrad | Sử dụng kích thước của gradient để điều chỉnh tốc độ học tập.  Ý tưởng của Adagrad là bắt đầu từ một điểm ngẫu nhiên trong không gian tham số của mô hình. Sau đó, phương pháp này sẽ sử dụng đạo hàm của hàm mục tiêu tại điểm hiện tại để tính toán hướng đi cần thiết để giảm thiểu hàm mục tiêu. Tuy nhiên, thay vì cập nhật các tham số của mô hình theo hướng đi này ngay lập tức, phương pháp Adagrad sẽ sử dụng một trọng lượng để pha trộn hướng đi này với hướng đi trước đó. Trọng lượng này sẽ được điều chỉnh dựa trên kích thước của gradient. | - Giúp mô hình vượt qua các điểm tối ưu cục bộ.  - Học nhanh hơn SGD.  - Ít bị ảnh hưởng bởi nhiễu trong dữ liệu. | - Có thể dẫn đến dao động trong quá trình tối ưu hóa.  - Khó điều chỉnh các tham số. |
| RMSProp (Root Mean Squared Prop) | Tương tự như Adagrad, nhưng sử dụng giá trị trung bình bình phương của gradient để điều chỉnh tốc độ học tập.  Ý tưởng của RMSProp là bắt đầu từ một điểm ngẫu nhiên trong không gian tham số của mô hình. Sau đó, phương pháp này sẽ sử dụng đạo hàm của hàm mục tiêu tại điểm hiện tại để tính toán hướng đi cần thiết để giảm thiểu hàm mục tiêu. Tuy nhiên, thay vì cập nhật các tham số của mô hình theo hướng đi này ngay lập tức, phương pháp RMSProp sẽ sử dụng một trọng lượng để pha trộn hướng đi này với hướng đi trước đó. Trọng lượng này sẽ được điều chỉnh dựa trên giá trị trung bình bình phương của gradient. | - Giúp mô hình vượt qua các điểm tối ưu cục bộ.  - Giảm thiểu khả năng quá khớp.  - Hiệu quả khi huấn luyện các mô hình lớn. | Khó điều chỉnh các tham số. |
| Adam (Adaptive Moment Estimation) | Kết hợp các ý tưởng của momentum và RMSProp để đạt được tốc độ học tập nhanh và ổn định hơn.  Ý tưởng của Adam là bắt đầu từ một điểm ngẫu nhiên trong không gian tham số của mô hình. Sau đó, phương pháp này sẽ sử dụng đạo hàm của hàm mục tiêu tại điểm hiện tại để tính toán hướng đi cần thiết để giảm thiểu hàm mục tiêu. Tuy nhiên, thay vì cập nhật các tham số của mô hình theo hướng đi này ngay lập tức, phương pháp Adam sẽ sử dụng một trọng lượng để pha trộn hướng đi này với hướng đi trước đó. Trọng lượng này sẽ được điều chỉnh dựa trên giá trị trung bình bình phương của gradient. | - Học nhanh hơn SGD.  - Giảm thiểu khả năng quá khớp.  - Hiệu quả khi huấn luyện các mô hình lớn. | Khó điều chỉnh các tham số. |
| AdaDelta | Tương tự như Adam, nhưng sử dụng giá trị trung bình của gradient và giá trị trung bình bình phương của gradient để điều chỉnh tốc độ học tập. Tuy nhiên, Adadelta có một số khác biệt so với Adam, bao gồm:  - Adadelta sử dụng giá trị trung bình của gradient và giá trị trung bình bình phương của gradient để điều chỉnh tốc độ học tập theo từng tham số riêng lẻ.  - Adadelta không sử dụng tham số β2, tham số này được sử dụng trong Adam để điều chỉnh tốc độ giảm dần của giá trị trung bình bình phương của gradient.  Ý tưởng của Adadelta là bắt đầu từ một điểm ngẫu nhiên trong không gian tham số của mô hình. Sau đó, phương pháp này sẽ sử dụng đạo hàm của hàm mục tiêu tại điểm hiện tại để tính toán hướng đi cần thiết để giảm thiểu hàm mục tiêu. Tuy nhiên, thay vì cập nhật các tham số của mô hình theo hướng đi này ngay lập tức, phương pháp Adadelta sẽ sử dụng một trọng lượng để pha trộn hướng đi này với hướng đi trước đó. Trọng lượng này sẽ được điều chỉnh dựa trên giá trị trung bình của gradient và giá trị trung bình bình phương của gradient. | - Học nhanh hơn SGD.  - Giảm thiểu khả năng quá khớp.  - Hiệu quả khi huấn luyện các mô hình lớn. | - Khó điều chỉnh các tham số. |

Bảng 1.1 So sánh các phương pháp optimizer

Chọn thuật toán tối ưu hóa phù hợp phụ thuộc vào nhiều yếu tố, bao gồm loại mô hình học máy đang được sử dụng, kích thước của tập huấn luyện và các yêu cầu hiệu suất.

# CONTINUAL LEARNING VÀ TEST PRODUCTION

## Continual Learning

Continual Learning (Học máy liên tục) là một nhánh của học máy tập trung vào việc phát triển các mô hình có thể học hỏi và thích ứng với dữ liệu mới trong quá trình sử dụng. Các mô hình học máy truyền thống thường được đào tạo trên một tập dữ liệu cố định, và sau đó được triển khai và sử dụng mà không cần cập nhật thêm. Tuy nhiên, trong nhiều trường hợp, dữ liệu thực tế có thể thay đổi theo thời gian, dẫn đến việc mô hình không còn phù hợp với dữ liệu mới. Continual Learning nhằm giải quyết vấn đề này bằng cách cho phép mô hình tiếp tục học hỏi từ dữ liệu mới mà không làm mất kiến thức đã học trước đó.

Trong các ứng dụng thực tế, dữ liệu luôn thay đổi theo thời gian. Ví dụ, trong các hệ thống nhận dạng khuôn mặt, các khuôn mặt mới luôn xuất hiện. Trong các hệ thống phân loại văn bản, các từ và cụm từ mới luôn được tạo ra. Nếu các mô hình học máy không thể học tập và thích ứng với dữ liệu mới, chúng sẽ dần trở nên lỗi thời và không còn hiệu quả.

Continual Learning có thể được áp dụng cho nhiều loại bài toán học máy khác nhau, bao gồm:

- Nhận dạng hình ảnh.

- Phân loại văn bản.

- Phân tích cảm xúc.

- Hệ thống khuyến nghị.

- Hệ thống điều khiển.

Có nhiều phương pháp Continual Learning khác nhau, mỗi phương pháp có những ưu nhược điểm riêng. Một số phương pháp phổ biến bao gồm:

- Forgetting prevention: Kỹ thuật này nhằm ngăn chặn mô hình quên kiến thức đã học trước đó. Một cách để làm điều này là sử dụng một bộ nhớ ngắn hạn để lưu trữ các thông tin quan trọng. Ví dụ, một mô hình phân loại hình ảnh có thể sử dụng một bộ nhớ ngắn hạn để lưu trữ các đặc điểm của các hình ảnh đã được phân loại chính xác.

- Data augmentation: Kỹ thuật này nhằm tăng cường tập dữ liệu hiện có bằng cách tạo ra các dữ liệu mới từ dữ liệu cũ. Điều này có thể giúp mô hình học hỏi từ nhiều dữ liệu hơn và tránh quên kiến thức đã học trước đó. Ví dụ, một mô hình phân loại hình ảnh có thể sử dụng các biến đổi hình ảnh để tạo ra các hình ảnh mới từ các hình ảnh cũ, chẳng hạn như xoay, lật, hoặc thay đổi độ sáng.

- Regularization: Kỹ thuật này nhằm hạn chế sự thay đổi của mô hình khi học hỏi dữ liệu mới. Điều này có thể giúp mô hình tránh học quá nhiều từ dữ liệu mới và quên kiến thức đã học trước đó. Ví dụ, một mô hình phân loại hình ảnh có thể sử dụng các hàm chi phí phi tuyến để ngăn chặn mô hình học quá nhiều từ dữ liệu mới.

## Test Production

Test Production là một quá trình kiểm tra các mô hình học máy trong môi trường sản xuất. Quá trình này thường được thực hiện theo các bước sau:

1. Training: Mô hình học máy được huấn luyện trên một tập dữ liệu đào tạo.

2. Evaluation: Mô hình học máy được đánh giá trên một tập dữ liệu đánh giá.

3. Deployment: Mô hình học máy được triển khai trong môi trường sản xuất.

Trong quá trình test production, mô hình học máy có thể gặp phải các vấn đề sau:

- Overfitting: Mô hình quá khớp với tập dữ liệu đào tạo, dẫn đến hiệu suất kém trên tập dữ liệu đánh giá và môi trường sản xuất.

- Underfitting: Mô hình không khớp với tập dữ liệu đào tạo, dẫn đến hiệu suất kém trên cả tập dữ liệu đánh giá và môi trường sản xuất.

- Data drift: Dữ liệu trong môi trường sản xuất khác với dữ liệu trong tập dữ liệu đào tạo, dẫn đến hiệu suất kém của mô hình.

Để giải quyết các vấn đề này, cần thực hiện các kỹ thuật sau:

- Data augmentation: Tạo thêm dữ liệu từ tập dữ liệu đào tạo để giảm thiểu overfitting.

- Regularization: Áp dụng các kỹ thuật regularizer để giảm thiểu overfitting.

- Ensembling: Kết hợp kết quả của nhiều mô hình để cải thiện hiệu suất.

- Data drift detection: Phát hiện sự thay đổi của dữ liệu trong môi trường sản xuất để có các biện pháp thích hợp.

## Ứng dụng của Continual Learning và Test Production

Continual Learning và Test Production có thể được ứng dụng trong nhiều bài toán khác nhau, bao gồm:

- Phân loại hình ảnh: Các mô hình phân loại hình ảnh có thể được sử dụng để phân loại các hình ảnh sản phẩm, các hình ảnh vi phạm bản quyền, hoặc các hình ảnh đe dọa an ninh. Trong những trường hợp này, dữ liệu có thể thay đổi theo thời gian, do các sản phẩm mới được thêm vào, các vi phạm bản quyền mới xuất hiện, hoặc các mối đe dọa an ninh mới. Continual Learning có thể được sử dụng để cập nhật các mô hình này với dữ liệu mới, giúp chúng luôn phù hợp với dữ liệu thực tế.

- Dự đoán nhu cầu khách hàng: Các mô hình dự đoán nhu cầu khách hàng có thể được sử dụng để dự đoán nhu cầu của khách hàng đối với các sản phẩm hoặc dịch vụ. Trong những trường hợp này, nhu cầu của khách hàng có thể thay đổi theo thời gian, do các yếu tố như sự thay đổi trong thị trường, các sự kiện như lễ hội hoặc ngày lễ, hoặc các chiến dịch tiếp thị mới. Continual Learning có thể được sử dụng để cập nhật các mô hình này với dữ liệu mới, giúp chúng dự đoán nhu cầu của khách hàng chính xác hơn.

- Hệ thống khuyến nghị: Các hệ thống khuyến nghị có thể được sử dụng để đề xuất các sản phẩm hoặc dịch vụ cho khách hàng. Trong những trường hợp này, dữ liệu về sở thích của khách hàng có thể thay đổi theo thời gian, do khách hàng mua sắm nhiều hơn, hoặc do các sự kiện như sinh nhật hoặc ngày kỷ niệm. Continual Learning có thể được sử dụng để cập nhật các hệ thống này với dữ liệu mới, giúp chúng đề xuất các sản phẩm hoặc dịch vụ phù hợp hơn với sở thích của khách hàng.

# TÀI LIỆU THAM KHẢO

Tiếng Việt

<https://viblo.asia/p/optimizer-hieu-sau-ve-cac-thuat-toan-toi-uu-gdsgdadam-Qbq5QQ9E5D8>

Tiếng Anh

<https://sebastianraschka.com/faq/docs/gradient-optimization.html>

<https://www.oreilly.com/library/view/designing-machine-learning/9781098107956/ch09.html>