TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**NGÔ LÊ HOÀNG - 52100890**

**BÁO CÁO CUỐI KỲ**

**MACHINE LEARNING**

**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**



**NGÔ LÊ HOÀNG - 52100890**

**BÁO CÁO CUỐI KỲ**

**MACHINE LEARNING**

Người hướng dẫn

**TS. LÊ ANH CƯỜNG**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

**LỜI CẢM ƠN**

Để hoàn thành bài báo cáo này, em xin gửi lời cảm ơn chân thành đến Ban giám hiệu trường Đại Học Tôn Đức Thắng vì đã tạo điều kiện thuận lợi nhất cho sinh viên khi học tập.

Em xin chân thành cảm ơn giảng viên bộ môn - TS. Lê Anh Cường đã tận tình giảng dạy chi tiết, đầy đủ kiến thức để em vận dụng vào bài báo cáo này.

Do chưa có nhiều kinh nghiệm trong việc làm báo cáo và sự hạn chế về kiến thức nên bài báo cáo không tránh khỏi những thiếu sót. Rất mong nhận được những ý kiến, đóng góp và phê bình từ phía cô để bài báo cáo được hoàn thiện hơn.

Lời cuối cùng, em xin kính chúc cô sức khỏe, thành công trong cuộc sống.

*TP. Hồ Chí Minh, ngày 19 tháng 12 năm 2023.*

*Tác giả*

*Hoàng*

*Ngô Lê Hoàng*

*(Ký tên và ghi rõ họ tên)*

**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 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 19 tháng 12 năm 2023*

*Tác giả*

*Hoàng*

*Ngô Lê Hoàng*

*(Ký tên và ghi rõ họ tên)*

**NỘI DUNG**

**CHƯƠNG 1:**

**Trình bày một bài nghiên cứu, đánh giá của em 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.**

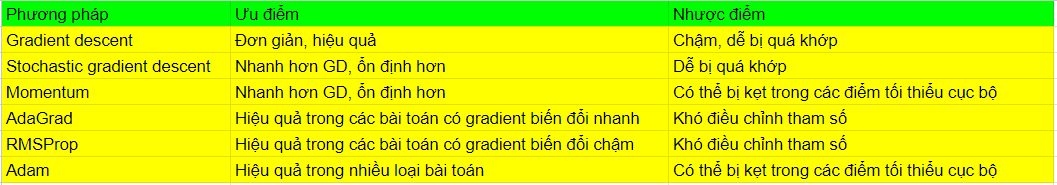
**1/ Tìm hiểu, so sánh các phương pháp Optimizer trong huấn luyện mô hình học máy;**

Trong huấn luyện mô hình học máy, thuật toán tối ưu hóa (optimizer) là một thành phần quan trọng giúp mô hình học được từ dữ liệu huấn luyện. Optimizer thực hiện việc cập nhật trọng số (weights) của mô hình theo một hướng dẫn nào đó để giảm thiểu hàm lỗi (loss function) của mô hình.

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

* Gradient descent (GD): Đây là phương pháp tối ưu hóa đơn giản và hiệu quả nhất. GD cập nhật trọng số theo hướng ngược lại của gradient của hàm lỗi.
* Stochastic gradient descent (SGD): Đây là một biến thể của GD, cập nhật trọng số dựa trên gradient của một ví dụ ngẫu nhiên từ tập huấn luyện. SGD có thể giúp mô hình học nhanh hơn GD, nhưng cũng dễ bị quá khớp (overfitting).
* Momentum: Momentum là một phương pháp cải tiến GD, sử dụng một momentum để tăng tốc độ cập nhật trọng số. Momentum có thể giúp mô hình học nhanh hơn và ổn định hơn GD.
* AdaGrad: AdaGrad là một phương pháp tối ưu hóa dựa trên ý tưởng giảm dần tốc độ cập nhật trọng số theo mức độ thay đổi của gradient. AdaGrad có thể giúp mô hình học hiệu quả hơn trong các bài toán có gradient biến đổi nhanh.
* RMSProp: RMSProp là một phương pháp tối ưu hóa tương tự AdaGrad, nhưng sử dụng một hàm mũ để giảm dần tốc độ cập nhật trọng số. RMSProp có thể giúp mô hình học hiệu quả hơn trong các bài toán có gradient biến đổi chậm.
* Adam: Adam là một phương pháp tối ưu hóa kết hợp các ưu điểm của Momentum và RMSProp. Adam có thể học hiệu quả trong nhiều loại bài toán khác nhau.

Dưới đây là bảng so sánh các phương pháp tối ưu hóa phổ biến:



Việc lựa chọn phương pháp tối ưu hóa phù hợp phụ thuộc vào nhiều yếu tố, bao gồm:

* Kiểu bài toán học máy
* Kích thước của tập huấn luyện
* Tính chất của hàm lỗi
* Sức mạnh tính toán của máy tính

Nhìn chung, Adam là một phương pháp tối ưu hóa hiệu quả và ổn định, có thể được sử dụng cho nhiều loại bài toán học máy. Tuy nhiên, Adam cũng có thể bị kẹt trong các điểm tối thiểu cục bộ trong một số trường hợp. Nếu gặp phải vấn đề này, có thể thử sử dụng các phương pháp khác như RMSProp hoặc AdaGrad.

Dưới đây là một số mẹo để lựa chọn và sử dụng phương pháp tối ưu hóa hiệu quả:

* Sử dụng phương pháp tối ưu hóa đơn giản cho các bài toán đơn giản. Đối với các bài toán đơn giản, có thể sử dụng các phương pháp tối ưu hóa đơn giản như GD hoặc SGD. Các phương pháp này dễ hiểu và triển khai, đồng thời cũng hiệu quả trong nhiều trường hợp.
* Sử dụng phương pháp tối ưu hóa phức tạp cho các bài toán phức tạp. Đối với các bài toán phức tạp, có thể sử dụng các phương pháp tối ưu hóa phức tạp hơn như Momentum, AdaGrad, RMSProp hoặc Adam. Các phương pháp này có thể giúp mô hình học hiệu quả hơn trong các bài toán này.
* Thử nghiệm với các phương pháp tối ưu hóa khác nhau. Trong một số trường hợp, các phương pháp tối ưu hóa khác nhau có thể cho kết quả khác nhau. Do đó, có thể thử nghiệm với các phương pháp tối ưu hóa khác nhau để tìm ra phương pháp phù hợp nhất cho bài toán cụ thể.

**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.**

Continual Learning (CL) là một lĩnh vực của học máy tập trung vào việc phát triển các mô hình học máy có thể học hỏi và thích nghi với dữ liệu mới mà không bị quên đi những gì đã học trước đó. Điều này rất quan trọng trong thế giới thực, nơi dữ liệu luôn thay đổi.

Khi xây dựng một giải pháp học máy để giải quyết một bài toán nào đó, cần xem xét khả năng sử dụng CL để cải thiện hiệu suất của giải pháp. CL có thể mang lại nhiều lợi ích cho các giải pháp học máy, bao gồm:

* Giảm thiểu lỗi: CL có thể giúp các mô hình học hỏi các mẫu mới mà không làm suy giảm hiệu suất trên các mẫu cũ. Điều này có thể giúp giảm thiểu lỗi trong các ứng dụng thực tế.
* Tăng cường hiệu suất: CL có thể giúp các mô hình học hỏi các mẫu mới để cải thiện hiệu suất. Điều này có thể giúp các giải pháp học máy hoạt động tốt hơn trong môi trường thực tế.
* Tăng cường khả năng thích ứng: CL có thể giúp các mô hình thích ứng với các thay đổi trong dữ liệu. Điều này có thể giúp các giải pháp học máy hoạt động tốt hơn trong thế giới thực, nơi dữ liệu luôn thay đổi.

Tuy nhiên, CL cũng có một số thách thức cần được xem xét, bao gồm:

* Hiệu suất: CL có thể làm giảm hiệu suất của mô hình trên các mẫu cũ.
* Tính phức tạp: CL có thể làm tăng độ phức tạp của mô hình và quá trình huấn luyện.
* Dữ liệu: CL đòi hỏi một lượng lớn dữ liệu mới để học hỏi.
* Để tận dụng lợi ích của CL và giảm thiểu các thách thức, cần cân nhắc một số yếu tố sau khi xây dựng giải pháp học máy:
* Loại bài toán: Một số bài toán phù hợp hơn với CL hơn các bài toán khác. Ví dụ, các bài toán phân loại hình ảnh có thể phù hợp hơn với CL hơn các bài toán phân loại văn bản.
* Kích thước và chất lượng dữ liệu: CL đòi hỏi một lượng lớn dữ liệu mới để học hỏi. Nếu dữ liệu mới không có sẵn, thì CL có thể không hiệu quả.
* Yêu cầu về hiệu suất: Nếu giải pháp học máy yêu cầu hiệu suất cao trên các mẫu cũ, thì cần phải cân nhắc kỹ lưỡng các kỹ thuật CL để tránh làm giảm hiệu suất.

Một số kỹ thuật CL phổ biến bao gồm:

* Incremental learning: Incremental learning là một cách tiếp cận trong đó các mô hình được cập nhật từng chút một với dữ liệu mới. Điều này có thể giúp các mô hình học hỏi các mẫu mới mà không làm xáo trộn các mẫu đã học trước đó.
* Lifelong learning: Lifelong learning là một cách tiếp cận trong đó các mô hình được học hỏi liên tục với dữ liệu mới. Điều này có thể giúp các mô hình thích ứng với các thay đổi trong dữ liệu.
* Online learning: Online learning là một cách tiếp cận trong đó các mô hình được học hỏi từ dữ liệu mới khi dữ liệu mới được tạo ra. Điều này có thể giúp các mô hình hoạt động tốt hơn trong môi trường thực tế, nơi dữ liệu luôn thay đổi.

Việc lựa chọn kỹ thuật CL phù hợp phụ thuộc vào các yếu tố như loại bài toán, kích thước và chất lượng dữ liệu, và yêu cầu về hiệu suất.

Dưới đây là một số ví dụ về ứng dụng CL trong thực tế:

* Phân loại hình ảnh: Các mô hình phân loại hình ảnh có thể được sử dụng để phân loại các loại hình ảnh mới, chẳng hạn như các loại hình ảnh mới xuất hiện trên mạng xã hội.
* Xem xét tín dụng: Các mô hình xem xét tín dụng có thể được sử dụng để đánh giá các rủi ro tín dụng mới, chẳng hạn như các rủi ro tín dụng mới xuất hiện do sự thay đổi trong thị trường kinh tế.
* Phát hiện gian lận: Các mô hình phát hiện gian lận có thể được sử dụng để phát hiện các loại gian lận mới, chẳng hạn như các loại gian lận mới xuất hiện do sự phát triển của công nghệ.

Với sự phát triển của công nghệ, dữ liệu luôn thay đổi. CL là một lĩnh vực quan trọng của học máy có thể giúp các giải pháp học máy hoạt động tốt hơn trong môi trường thực tế.

**Test Production**

Test Production là một quá trình trong đó một mô hình học máy được triển khai trong môi trường sản xuất trong khi vẫn đang được thử nghiệm. Điều này cho phép các nhà phát triển học hỏi về hiệu suất của mô hình trong môi trường thực tế và thực hiện các thay đổi cần thiết để cải thiện hiệu suất.

*Lợi ích của Test Production*

Test Production có thể mang lại nhiều lợi ích cho các giải pháp học máy, bao gồm:

* Giảm thiểu lỗi: Test Production có thể giúp các nhà phát triển phát hiện các lỗi trong mô hình trước khi mô hình được triển khai rộng rãi. Điều này có thể giúp giảm thiểu lỗi trong các ứng dụng thực tế.
* Tăng cường hiệu suất: Test Production có thể giúp các nhà phát triển cải thiện hiệu suất của mô hình bằng cách thực hiện các thay đổi cần thiết dựa trên kết quả thử nghiệm.
* Tăng cường khả năng thích ứng: Test Production có thể giúp các nhà phát triển cải thiện khả năng thích ứng của mô hình với các thay đổi trong môi trường thực tế.

*Thách thức của Test Production*

Test Production cũng có một số thách thức cần được xem xét, bao gồm:

* Quản lý: Test Production có thể phức tạp để quản lý, vì cần phải có một quy trình rõ ràng để triển khai và kiểm tra mô hình trong môi trường sản xuất.
* Bảo mật: Test Production có thể gây ra các rủi ro bảo mật, vì mô hình có thể được triển khai trong môi trường sản xuất trong khi vẫn chưa được hoàn thiện.

*Các bước thực hiện Test Production*

Có thể thực hiện Test Production theo các bước sau:

* Xác định phạm vi thử nghiệm: Trước khi bắt đầu thử nghiệm, cần xác định phạm vi thử nghiệm. Điều này bao gồm việc xác định các loại lỗi mà cần phải phát hiện, các chỉ số hiệu suất cần được đo lường, và các thay đổi cần được thực hiện dựa trên kết quả thử nghiệm.
* Thiết kế thử nghiệm: Sau khi xác định phạm vi thử nghiệm, cần thiết kế thử nghiệm. Điều này bao gồm việc xác định các trường hợp thử nghiệm, các kịch bản thử nghiệm, và các tiêu chí chấp nhận.
* Triển khai mô hình trong môi trường sản xuất: Sau khi thiết kế thử nghiệm, cần triển khai mô hình trong môi trường sản xuất. Điều này có thể được thực hiện bằng cách sử dụng một hệ thống canary release để triển khai mô hình cho một nhóm nhỏ người dùng trước khi triển khai cho tất cả người dùng.
* Chạy thử nghiệm: Sau khi triển khai mô hình trong môi trường sản xuất, cần chạy thử nghiệm để thu thập dữ liệu. Dữ liệu này sẽ được sử dụng để đánh giá hiệu suất của mô hình và thực hiện các thay đổi cần thiết.
* Đánh giá kết quả thử nghiệm: Sau khi chạy thử nghiệm, cần đánh giá kết quả thử nghiệm. Điều này bao gồm việc xác định các lỗi cần được sửa chữa, các chỉ số hiệu suất cần được cải thiện, và các thay đổi cần được thực hiện đối với mô hình.
* Thực hiện các thay đổi: Sau khi đánh giá kết quả thử nghiệm, cần thực hiện các thay đổi đối với mô hình. Điều này có thể bao gồm việc sửa chữa lỗi, cải thiện hiệu suất, hoặc điều chỉnh mô hình để thích ứng với các thay đổi trong môi trường thực tế.
* Lặp lại các bước từ 2 đến 6: Quá trình Test Production có thể được lặp lại nhiều lần để cải thiện hiệu suất của mô hình.

Một ví dụ về Test Production là việc triển khai một mô hình phân loại hình ảnh trong một ứng dụng nhận dạng khuôn mặt. Trong trường hợp này, các nhà phát triển có thể sử dụng Test Production để phát hiện các lỗi trong mô hình, chẳng hạn như các lỗi phân loại khuôn mặt sai. Các nhà phát triển cũng có thể sử dụng Test Production để cải thiện hiệu suất của mô hình, chẳng hạn như giảm thiểu số lần phân loại sai.