

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



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

DỰ ÁN CUỐI KÌ

Người hướng dẫn: **GV LÊ ANH CƯỜNG**

Người thực hiện: **CHÂU VI HOÀNG LONG – 51800890**

Lớp : 18050302

Khoá : 22

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**



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

DỰ ÁN CUỐI KÌ

Người hướng dẫn: **GV LÊ ANH CƯỜNG**

Người thực hiện: **CHÂU VI HOÀNG LONG – 51800890**

Lớp : **18050302**

Khoá : **22**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023

LỜI CẢM ƠN

Em xin chân thành cảm ơn thầy Lê Anh Cường đã nhiệt tình giúp đỡ em trong quá trình học tập trên lớp và trong suốt quá trình làm bài tập này. Chúc thầy luôn giữ được ngọn lửa nhiệt huyết trên giảng đường, một lần nữa em xin chân thành cảm ơn thầy.

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)

Châu Vi Hoàng Long

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

Ở dự án cuối kì này. Em sẽ trình bày một bài nghiên cứu, đánh giá của em về các vấn đề sau:

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

LỜI CẢM ƠN	i
PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN	iii
TÓM TẮT	iv
MỤC LỤC	1
CHƯƠNG 1 – TÌM HIỂU, SO SÁNH CÁC PHƯƠNG PHÁP OPTIMIZER TRONG HUẤN LUYỆN MÔ HÌNH HỌC MÁY	3
1.1 Tìm hiểu các phương pháp Optimizer.	3
1.1.1 Optimizer là gì?	3
1.1.2 Các phương pháp Optimizer.	3
1.2. So sánh	5
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.	8
2.1. Continual Learning.	8
2.2. Test Production	9
TÀI LIỆU THAM KHẢO	11

DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT

CÁC KÝ HIỆU

CÁC CHỮ VIẾT TẮT

GD Gradient Descent

SGD Stochastic Gradient Descent

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

1.1 Tìm hiểu các phương pháp Optimizer.

1.1.1 Optimizer là gì?

- Trong huấn luyện mô hình học máy, thuật toán tối ưu hóa (optimizer) là một phần quan trọng trong quá trình tìm ra các tham số tối ưu cho mô hình. Thuật toán tối ưu hóa sẽ sử dụng hàm mất mát (loss function) để tìm cách giảm thiểu hàm này, từ đó cải thiện hiệu suất của mô hình.

1.1.2 Các phương pháp Optimizer.

- 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 trong huấn luyện mô hình học máy bao gồm:
 - **Gradient Descent (GD):** là phương pháp tối ưu hóa đơn giản và dễ hiểu, hoạt động bằng cách cập nhật các tham số của mô hình theo hướng ngược lại với hướng gradient của hàm mất mát. GD có thể dễ dàng triển khai, thường hoạt động tốt trên các bài toán có hàm mất mát liên tục và có hiệu quả trong nhiều trường hợp. Tuy nhiên, GD có thể gặp phải vấn đề hội tụ chậm, đặc biệt là khi hàm mất mát có nhiều đỉnh hoặc khi tập dữ liệu huấn luyện lớn.
 - **Stochastic Gradient Descent (SGD):** là một biến thể của GD, trong đó các tham số của mô hình được cập nhật sau mỗi mẫu dữ liệu trong tập huấn luyện. SGD thường nhanh hơn GD và yêu cầu ít bộ nhớ hơn, đặc biệt là khi làm việc với tập dữ liệu lớn. Nhưng có thể dẫn đến việc hội tụ chậm hơn trong một số trường hợp và khó đạt được sự ổn

đỉnh trong quá trình hội tụ. Ngoài ra, điểm tối ưu đạt được có thể không tốt như GD do tính ngẫu nhiên của mẫu.

- **Momentum:** là một phương pháp tối ưu hóa cải tiến GD, bằng cách sử dụng momentum để tăng tốc độ hội tụ. Momentum hoạt động bằng cách tích lũy gradient của các bước trước đó và sử dụng tổng gradient này để cập nhật các tham số của mô hình. Điều này giúp vượt qua các điểm yên ngựa và tránh bị kẹt tại các điểm tối ưu cục bộ, giúp giảm độ dao động và giúp quá trình tối ưu hóa ổn định hơn. Momentum có thể giúp tăng tốc độ hội tụ của GD, nhưng cũng có thể dẫn đến việc vượt qua đỉnh của hàm mất mát, nếu được cài đặt không đúng, nó có thể vượt quá điểm tối ưu và gây ra quá tối ưu dẫn đến độ lỗi tăng lên.
- **Adagrad:** là một phương pháp tối ưu hóa cải tiến GD, bằng cách điều chỉnh tốc độ học tập theo độ lớn của gradient. Adagrad hoạt động bằng cách lưu trữ các bình phương của gradient của các bước trước đó. Sau đó, tốc độ học tập sẽ được giảm dần theo độ lớn của các bình phương gradient này. Điều này giúp giảm tỷ lệ học tập cho các tham số có gradient lớn và tăng khả năng hội tụ. Adagrad có thể giúp cải thiện hiệu quả của GD, Tuy nhiên, Adagrad có thể gặp vấn đề của việc giảm tỷ lệ học tập quá nhanh theo thời gian, dẫn đến việc quá tối ưu hoặc có thể gặp phải vấn đề hội tụ chậm.
- **Root Mean Square Propagation (RMSProp):** là một phương pháp tối ưu hóa cải tiến Adagrad, bằng cách sử dụng bình phương trung bình của gradient của các bước trước đó để điều chỉnh tốc độ học tập. Điều này giúp giảm tỷ lệ học tập cho các tham số có gradient lớn và đạt được sự ổn định hơn trong quá trình huấn luyện. RMSProp hoạt động tương tự như Adagrad, nhưng có thể hiệu quả hơn trong một số trường hợp. Tuy nhiên, RMSProp không có cơ chế tự điều chỉnh tỷ lệ

học tập như Adam. Điều này có thể yêu cầu việc thay đổi tỷ lệ học tập thủ công để đạt hiệu suất tốt nhất trên từng bài toán cụ thể. Ngoài ra, RMSProp có thể yêu cầu điều chỉnh các siêu tham số như tỷ lệ học tập và hệ số giảm động của gradient.

- **Adam:** là một phương pháp tối ưu hóa cải tiến Adagrad và RMSProp, bằng cách kết hợp các ưu điểm của cả hai phương pháp này. Adam hoạt động bằng cách sử dụng cả bình phương gradient và bình phương trung bình của gradient của các bước trước đó để điều chỉnh tốc độ học tập, giúp tăng khả năng hội tụ và ổn định trong quá trình huấn luyện. Adam là một phương pháp tối ưu hóa hiệu quả và được sử dụng rộng rãi trong các ứng dụng học máy. Tuy nhiên, Adam có thể yêu cầu điều chỉnh các siêu tham số như tỷ lệ học tập và độ ngược của momentum để đạt hiệu suất tốt nhất. Ngoài ra, trong một số trường hợp, Adam có thể dẫn đến quá tối ưu hoặc không thể hội tụ tốt.
- Việc lựa chọn phương pháp tối ưu hóa thích hợp phụ thuộc vào nhiều yếu tố, bao gồm:
 - Kích thước của tập dữ liệu huấn luyện: Nếu tập dữ liệu huấn luyện lớn, thì SGD hoặc Adam có thể là lựa chọn tốt hơn GD.
 - Hình dạng của hàm mất mát: Nếu hàm mất mát có nhiều đỉnh hoặc khi tập dữ liệu huấn luyện có nhiễu, thì các phương pháp tối ưu hóa sử dụng momentum hoặc bình phương gradient có thể hiệu quả hơn.
 - Kiến trúc của mô hình: Các mô hình phức tạp có thể yêu cầu các phương pháp tối ưu hóa hiệu quả hơn.

1.2. So sánh

Phương pháp	Ưu điểm	Nhược điểm
Gradient Descent (GD)	<ul style="list-style-type: none"> - Đơn giản, dễ triển khai - Hoạt động tốt trên các bài toán có hàm mất mát liên tục - Hiệu quả trong nhiều trường hợp 	<ul style="list-style-type: none"> - Có thể hội tụ chậm khi hàm mất mát có nhiều đỉnh hoặc tập dữ liệu huấn luyện quá lớn
Stochastic Gradient Descent (SGD)	<ul style="list-style-type: none"> - Nhanh hơn GD - Yêu cầu ít bộ nhớ khi làm việc với tập dữ liệu lớn 	<ul style="list-style-type: none"> - Có thể hội tụ chậm hơn trong một số trường hợp - Điểm tối ưu không tốt do tính ngẫu nhiên của mẫu
Momentum	<ul style="list-style-type: none"> - Tăng tốc độ hội tụ - Tránh bị kẹt tại các điểm tối ưu cục bộ giúp quá trình tối ưu hóa ổn định hơn 	<ul style="list-style-type: none"> - Có thể dẫn đến việc vượt qua đỉnh của hàm mất mát - Có thể vượt quá điểm tối ưu, gây ra quá tối ưu dẫn đến độ lỗi tăng lên
Adagrad	<ul style="list-style-type: none"> - Điều chỉnh tốc độ học tập cho từng tham số dựa trên lịch sử gradient trước đó của từng tham số 	<ul style="list-style-type: none"> - Có thể gặp phải vấn đề của việc giảm tỷ lệ học tập quá nhanh theo thời gian, dẫn đến việc quá tối ưu hoặc hội tụ chậm.
RMSProp	<ul style="list-style-type: none"> - Tương tự như Adagrad, nhưng có thể hiệu quả hơn trong một số trường hợp - Giúp giảm tỷ lệ học tập 	<ul style="list-style-type: none"> - Không có cơ chế tự điều chỉnh tỷ lệ học tập - Có thể yêu cầu việc thay đổi tỷ lệ học tập thủ công

	cho các tham số có gradient lớn và đạt được sự ổn định hơn trong quá trình huấn luyện	để đạt hiệu suất tốt nhất
Adam	<ul style="list-style-type: none"> - Kết hợp các ưu điểm của Adagrad và RMSProp - Sử dụng cả bình phương gradient và bình phương trung bình của gradient của các bước trước đó để điều chỉnh tốc độ học tập, giúp tăng khả năng hội tụ và ổn định trong quá trình huấn luyện 	<ul style="list-style-type: none"> - Có thể yêu cầu điều chỉnh các siêu tham số như tỷ lệ học tập và độ ngược của momentum để đạt hiệu suất tốt nhất - Có thể quá tối ưu hoặc không thể hội tụ tốt.

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.

2.1. Continual Learning.

- Continual Learning là: một lĩnh vực của học máy tập trung vào việc phát triển các mô hình có thể học hỏi và thích ứng với những thay đổi trong dữ liệu đầu vào. Các mô hình học máy truyền thống thường được đào tạo trên một tập dữ liệu cố định và sau đó sẽ không thể cải thiện hiệu suất của chúng khi dữ liệu đầu vào thay đổi. Continual learning nhằm giải quyết vấn đề này bằng cách cho phép các mô hình tiếp tục học hỏi từ dữ liệu đầu vào mới mà không làm mất đi những gì chúng đã học được trước đó.
- Có nhiều cách tiếp cận khác nhau để xây dựng các giải pháp continual learning. Dưới đây là một số phương pháp continual learning phổ biến:
 - **Online Learning:** là một phương pháp học máy trong đó mô hình được đào tạo trên từng mẫu dữ liệu mới một cách liên tục. Điều này cho phép mô hình cập nhật kiến thức của mình một cách liên tục khi dữ liệu đầu vào thay đổi.
 - **Incremental Learning:** là một phương pháp học máy trong đó mô hình được đào tạo trên một tập dữ liệu mới dựa trên kiến thức mà nó đã học được từ tập dữ liệu cũ. Điều này cho phép mô hình tránh phải học lại những gì nó đã biết trước đó.
 - **Forgetting Prevention:** là một phương pháp nhằm ngăn chặn mô hình quên những gì nó đã học được trước đó. Điều này có thể được

thực hiện bằng cách sử dụng các kỹ thuật như elastic weight consolidation hoặc path dropout.

- **Data Augmentation:** là một phương pháp tạo ra các mẫu dữ liệu mới từ các mẫu dữ liệu cũ. Điều này có thể giúp mô hình học hỏi từ dữ liệu mới mà không làm mất đi những gì nó đã học được trước đó.

2.2. Test Production

- **Test production:** là một quá trình kiểm tra các giải pháp học máy trong môi trường sản xuất. Quá trình này nhằm đảm bảo rằng các giải pháp học máy hoạt động chính xác và hiệu quả trong môi trường thực tế.
- Khi xây dựng một giải pháp học máy để giải quyết một bài toán nào đó, cần phải cân nhắc đến cả việc kiểm tra trong môi trường sản xuất. Điều này là cần thiết để đảm bảo rằng giải pháp học máy sẽ hoạt động tốt trong môi trường thực tế, nơi có thể có các yếu tố khác nhau có thể ảnh hưởng đến hiệu suất của giải pháp.
- Một số lợi ích của việc thực hiện test production:
 - Đảm bảo chất lượng của giải pháp học máy
 - Phát hiện các lỗi hoặc vấn đề tiềm ẩn trong giải pháp học máy
 - Thu thập dữ liệu phản hồi từ người dùng để cải thiện giải pháp học máy
- Khi thực hiện test production, cần chú ý đến các yếu tố sau:

- **Chọn dữ liệu test:** Dữ liệu test cần được chọn cẩn thận để phản ánh các điều kiện trong môi trường sản xuất. Dữ liệu test có thể được thu thập từ môi trường sản xuất hoặc được tạo ra bằng cách sử dụng các kỹ thuật như data augmentation.
- **Chọn các tiêu chí đánh giá:** Cần xác định các tiêu chí đánh giá để đánh giá hiệu suất của giải pháp học máy trong môi trường sản xuất. Các tiêu chí đánh giá có thể bao gồm độ chính xác, độ tin cậy, tốc độ và khả năng mở rộng.
- **Cách thức thực hiện test:** Cần xác định cách thức thực hiện test để đảm bảo rằng các kết quả test là chính xác và đáng tin cậy. Các phương pháp test phổ biến bao gồm kiểm tra thủ công, kiểm tra tự động và kiểm tra A/B.
- Một số lưu ý khi thực hiện test production
 - Test production cần được thực hiện thường xuyên, ngay cả sau khi giải pháp học máy đã được triển khai. Điều này là cần thiết để đảm bảo rằng giải pháp học máy vẫn hoạt động tốt trong môi trường thực tế, nơi có thể có các thay đổi trong dữ liệu hoặc yêu cầu.
 - Test production cần được thực hiện bởi một nhóm kỹ sư hoặc người dùng có kinh nghiệm. Nhóm này cần hiểu rõ về giải pháp học máy và môi trường sản xuất nơi giải pháp học máy sẽ được sử dụng.
 - Các kết quả test cần được phân tích cẩn thận để xác định bất kỳ vấn đề tiềm ẩn nào. Nếu có bất kỳ vấn đề nào được phát hiện, thì cần phải thực hiện các biện pháp khắc phục để đảm bảo rằng giải pháp học máy hoạt động tốt.

TÀI LIỆU THAM KHẢO

- i. <https://viblo.asia/p/optimizer-hieu-sau-ve-cac-thuat-toan-toi-uu-gdsgdadam-Qbq5QQ9E5D8>
- ii. Continual Learning: Foundations, Methods, and Applications by Christoph H. Lampert, Benjamin D. Horne, and Andreas Krause
- iii. A Survey on Production Testing of Machine Learning Models by Jialin Pan, Lei Yang, and Jie Yang