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**

**TÌM HIỂU VỀ OPTIMIZER, CONTINUAL LEARNING VÀ TEST PRODUCTION, SO SÁNH CÁC PHƯƠNG PHÁP OPTIMIZER**

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

*Người thực hiện*: **NGUYỄN THÀNH HUY – 52100895**

Lớp **: 21050301**

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**



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

**TÌM HIỂU VỀ OPTIMIZER, CONTINUAL LEARNING VÀ TEST PRODUCTION, SO SÁNH CÁC PHƯƠNG PHÁP OPTIMIZER**

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

*Người thực hiện*: **NGUYỄN THÀNH HUY – 52100895**

Lớp **: 21050201**

Khoá  **: 25**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

LỜI CẢM ƠN

Đầu tiên, chúng em xin gửi lời cảm ơn chân thành đến Thầy TS Lê Anh Cường – Giảng viên khoa Công nghệ thông tin – Trường đại học Tôn Đức Thắng, đã hỗ trợ và giúp đỡ nhiệt tình trong quá trình thực hiện dự án này.

Chúng em trân trọng cảm ơn Thầy Cô giảng viên Trường đại học Tôn Đức Thắng nói chung cũng như Thầy Cô giảng viên khoa Công nghệ thông tin nói riêng đã giảng dạy và truyền đạt nhiều kinh nghiệm quý trong suốt quá trình học tập tại trường.

Cuối cùng, xin cám ơn gia đình, bạn bè đã luôn động viên và đồng hành trong quá trình học tập cũng như quá trình thực hiện Dự án này.

Mặc dù rất cẩn thận trong quá trình thực hiện dự án cũng như viết báo cáo nhưng cũng không tránh khỏi những thiếu sót. Em rất mong nhận được sự góp ý từ các Thầy/Cô để dự án được hoàn thiện hơn.

Xin chân thành cám ơn!

**DỰ ÁN ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Em xin cam đoan đây là công trình nghiên cứu của riêng em 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 báo cáo 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 có bất kỳ** **sự gian lận nào em xin hoàn toàn chịu trách nhiệm về nội dung báo cáo của mình**. Trường Đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do em 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)*

**

*Nguyễn Thành Huy*

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

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

Chương II: Tìm hiểu về Continual Learning và Test Production.

MỤC LỤC

[LỜI CẢM ƠN i](#_Toc154176736)

[TÓM TẮT v](#_Toc154176737)

[MỤC LỤC 1](#_Toc154176738)

[DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ 2](#_Toc154176739)

[DANH MỤC HÌNH 2](#_Toc154176740)

[DANH MỤC BẢNG 2](#_Toc154176741)

[DANH MỤC CÁC CHỮ VIẾT TẮT 3](#_Toc154176742)

[CHƯƠNG I: TÌM HIỂU, SO SÁNH CÁC PHƯƠNG PHÁP OPTIMIZER TRONG HỌC MÁY 4](#_Toc154176743)

[I. Optimizer là gì? 4](#_Toc154176744)

[1. Giới thiệu 4](#_Toc154176745)

[2. Các khái niệm 5](#_Toc154176746)

[3. Các phương pháp Optimizer phổ biến 5](#_Toc154176747)

[CHƯƠNG II: TÌM HIỂU VỀ CONTINUAL LEARNING VÀ TEST PRODUCTION 12](#_Toc154176748)

[I. Continual Learning 12](#_Toc154176749)

[1. Điểm khác biệt của Continual Learning so với các mô hình Machine Learning truyền thống 12](#_Toc154176750)

[2. Những thách thức của Continual Learning và giải pháp 14](#_Toc154176751)

[II. Test Production 18](#_Toc154176752)

[1. Test Production là gì? 18](#_Toc154176753)

[2. Các bước của quy trình test production 18](#_Toc154176754)

[3. Mục đích của Test Production 19](#_Toc154176755)

[TÀI LIỆU THAM KHẢO 21](#_Toc154176756)

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

## DANH MỤC HÌNH

## DANH MỤC BẢNG

DANH MỤC CÁC CHỮ VIẾT TẮT

SGD: Stochastic Gradient Descent

Adam: Adaptive Moment Estimation

RMSprop: Root Mean Square Propagation

CHƯƠNG I: TÌM HIỂU, SO SÁNH CÁC PHƯƠNG PHÁP OPTIMIZER TRONG HỌC MÁY

1. Optimizer là gì?
2. Giới thiệu

Trong học máy, Optimizer (hay còn được gọi là bộ tối ưu hóa) là một thành phần quan trọng của quá trình huấn luyện mô hình máy học. Bộ tối ưu hóa giúp tìm kiếm các tham số (parameters) tốt nhất cho mô hình dựa trên dữ liệu huấn luyện và hàm mất mát (loss function).

Mục tiêu của quá trình tối ưu hóa là tìm kiếm bộ tham số mô hình (như trọng số “weight” và bias) để mô hình đạt được hiệu suất tốt nhất trên dữ liệu huấn luyện. Điều này đòi hỏi cần phải làm giảm thiểu hàm mất mát, tức là khoảng cách giữa các dự đoán của mô hình và các giá trị thực tế.

Một optimizer hoạt động bằng cách điều chỉnh các trọng số (weights) và các tham số (parameters) của mô hình để giảm giá trị của hàm mất mát. Quá trình này được thực hiện thông qua việc cập nhật các trọng số theo hướng và khoảng độ lớn thích hợp bằng cách sử dụng các thuật toán optimizer khác nhau. Dưới dây là một số optimizer phổ biến trong học máy:

* Stochastic Gradient Descent (SGD)
* Adam (Adaptive Moment Estimation)
* RMSprop (Root Mean Square Propagation)

Mỗi thuật toán tối ưu hóa có những cách tiếp cận và đặc điểm riêng, nhưng chủ yếu đều dựa trên việc điều chỉnh các tham số mô hình dựa trên đạo hàm của hàm mất mát và một tốc độ học (learning rate) để đạt được sự hội tụ tối ưu. Các optimizer có thể có ảnh hưởng đáng kể đến hiệu suất và tốc độ hội tụ của mô hình trong quá trình huấn luyện. Việc lựa chọn optimizer thường phụ thuộc vào đặc tính của bài toán cụ thể và đặc điểm của dữ liệu.

1. Các khái niệm
2. Mini-batch

* Mini-batches là nhóm nhỏ các mẫu dữ liệu được lựa chọn ngẫu nhiên từ tập dữ liệu huấn luyện.
* Kích thước của mini-batch là một tham số quan trọng, ảnh hưởng đến hiệu suất và tốc độ của thuật toán.

1. Learning rate (η)

* Là tham số quyết định độ lớn của bước cập nhật trọng số.
* Quá lớn có thể dẫn đến dao động và không hội tụ, trong khi quá nhỏ có thể làm chậm quá trình học.

1. Các phương pháp Optimizer phổ biến
   1. Stochastic Gradient Descent (SGD)

Stochastic Gradient Descent (SGD) là một trong những phương pháp tối ưu hóa phổ biến được sử dụng trong quá trình huấn luyện mô hình machine learning.

1. Nguyên lý hoạt động

Bước 1: Khởi tạo ngẫu nhiên các trọng số của mô hình.

Bước 2: Dữ liệu huấn luyện được chia thành các mini-batches (nhóm nhỏ) ngẫu nhiên. Mỗi mini-batch chứa một số lượng nhỏ các mẫu dữ liệu.

Bước 3: Với mỗi mini-batch, thuật toán tính gradient của hàm mất mát (loss function) theo trọng số của mô hình.

Bước 4: Sử dụng gradient tính được, cập nhật trọng số theo hướng và khoảng độ lớn thích hợp để giảm giá trị của hàm mất mát. Công thức cập nhật trọng số được xác định bởi tỷ lệ học (learning rate).

Công thức cập nhật trọng số trong SGD:

Trong đó:

* là vector trọng số tại bước thứ *i*
* là gradient của hàm mất mát *J* tại với dữ liệu mini-batch
* là tỉ lệ học (learning rate)

Bước 5: Lặp lại bước 3 và bước 4 cho đến khi đạt được điều kiện dừng (số lượng lặp, giá trị mất mát đủ nhỏ,…)

Lợi ích của SGD đến từ việc sử dụng mini-batches, giúp giảm độ phức tạp tính toán so với Gradient Descent thông thường, đặc biệt là khi có một lượng lớn dữ liệu. SGD cũng giúp tìm kiếm không gian trọng số nhanh chóng hơn, nhưng có thể gặp vấn đề về dao động và nhiễu trong quá trình học.

1. Bài toán phù hợp

Dưới đây là một số tình huống mà SGD thường được ưa chuộng:

* Dữ liệu Lớn: SGD thích hợp cho các tập dữ liệu lớn, vì nó chỉ cần sử dụng một mini-batch nhỏ mỗi lần cập nhật, giảm bớt tải lớn về tài nguyên tính toán so với Gradient Descent thông thường.
* Phân loại và Hồi quy: SGD thường được sử dụng hiệu quả trong bài toán phân loại và hồi quy, đặc biệt là khi có nhiều điểm dữ liệu và mô hình cần được huấn luyện trên dữ liệu đó.
* Quá Trình Học Online: Khi dữ liệu được thêm vào liên tục, SGD là lựa chọn tốt cho quá trình học online, nơi mô hình cần được cập nhật khi có mẫu dữ liệu mới.
* Không Gian Trạng Thái Lớn: Khi không gian trạng thái của mô hình lớn và không khả dụng để lưu trữ toàn bộ tập dữ liệu trong bộ nhớ, SGD trở nên hữu ích vì nó chỉ yêu cầu một phần nhỏ của dữ liệu để tính toán gradient và cập nhật trọng số.
* Dữ Liệu Nhiễu: Khi dữ liệu có nhiễu, SGD có thể giúp tránh các vấn đề về quá mức điều chỉnh và giúp mô hình tự động thích ứng với các biến động ngẫu nhiên.
* Tối Ưu Hóa Đa Nhiệm: Trong các tình huống tối ưu hóa đa nhiệm, nơi một mô hình cần học từ nhiều nhiệm vụ khác nhau, SGD có thể được sử dụng để cập nhật trọng số của mô hình theo cách liên tục và linh hoạt.
  1. Adam (Adaptive Moment Estimation)

Adam (Adaptive Moment Estimation) là một phương pháp optimizer trong machine learning, được thiết kế để cải thiện quá trình tối ưu hóa bằng cách kết hợp cả các ưu điểm của phương pháp Momentum và RMSprop. Adam tự động điều chỉnh tỷ lệ học cho từng tham số của mô hình và duy trì một trung bình có độ đậm của gradient.

1. Nguyên lý hoạt động

Bước 1: Khởi tạo các tham số ban đầu bao gồm trọng số , các trung bình độ đậm và ban đầu là 0.

Bước 2: Tính gradient của hàm mất mát theo trọng số θ, ký hiệu là ​.

Bước 3: Cập nhật các trung bình độ đậm và theo các công thức sau:

* Trong đó, và là các hệ số giảm xuống, thường được đặt giá trị mặc định là 0.9 và 0.999

Bước 4: Tính ước lượng chính xác của độ đậm và để giảm hiệu ứng của bias khi bắt đầu (đặc biệt quan trọng khi và lớn)

Bước 5: Cập nhật trọng số θ bằng cách sử dụng các ước lượng chính xác của và , cũng như tỷ lệ học (η) và một giá trị nhỏ để tránh chia cho 0 (ϵ):

* Trong đó, là tỉ lệ học (learning rate), và là một giá trị rất nhỏ (thường là )

Bước 6: Lặp lại các bước trên cho mỗi epoch hoặc mini-batch cho đến khi đạt được mục tiêu dừng hoặc số lượng epoch đã đủ

Adam sử dụng các trung bình độ đậm để theo dõi sự thay đổi của gradient và bình phương gradient theo thời gian. Đồng thời, nó cũng có tính chất adaptive, tự động điều chỉnh tỷ lệ học cho từng tham số của mô hình. Nhờ vào các tính chất này, Adam thường đạt được hiệu suất tốt trên nhiều bài toán và kiến trúc mô hình.

1. Bài toán phù hợp

Optimizer Adam thường phù hợp và hiệu quả trong nhiều loại dữ liệu và bài toán machine learning. Dưới đây là một số trường hợp Adam thường được ưa chuộng:

* Dữ Liệu Lớn: Adam thường hoạt động tốt trên các tập dữ liệu lớn, nơi sự adaptive của nó trong việc điều chỉnh tỷ lệ học có thể giúp tối ưu hóa mô hình hiệu quả.
* Bài toán Phân loại và Hồi quy: Adam thường được sử dụng hiệu quả trong các bài toán phân loại và hồi quy, đặc biệt là khi có độ phức tạp cao và nhiều tham số.
* Mô Hình Học Sâu (Deep Learning): Adam thường là lựa chọn phổ biến trong deep learning, nơi sự tự điều chỉnh của nó giúp học tốt trên nhiều lớp và tham số.
* Tối Ưu Hóa Đa Nhiệm: Trong các tình huống tối ưu hóa đa nhiệm, Adam có thể hiệu quả khi mô hình cần học từ nhiều nhiệm vụ khác nhau.
* Dữ Liệu Nhiễu: Adam có thể ổn định quá trình học trên dữ liệu nhiễu, đặc biệt là khi dữ liệu chứa đựng độ nhiễu và biến động ngẫu nhiên.
* Quá Trình Học Online: Adam có thể phù hợp cho quá trình học online, nơi mô hình cần được cập nhật khi có dữ liệu mới được thêm vào.
* Bài toán với Đặc Trưng Đa Dạng: Adam thường hoạt động tốt khi đối mặt với các bài toán có đặc trưng đa dạng và phức tạp.
  1. RMSprop (Root Mean Square Propagation)

RMSprop (Root Mean Square Propagation) là một phương pháp tối ưu hóa được sử dụng trong quá trình huấn luyện mô hình học máy. Nó là một biến thể của phương pháp Gradient Descent được thiết kế để cải thiện hiệu suất tối ưu hóa bằng cách điều chỉnh tỉ lệ học (learning rate) dựa trên thông tin độ lớn của gradient.

1. Nguyên lý hoạt động

Bước 1: Khởi tạo tham số ban đầu bao gồm trọng số θ, và trung bình bình phương gradient ban đầu là 0.

Bước 2: Tính gradient của hàm mất mát theo trọng số θ, ký hiệu là

Bước 3: Cập nhật trung bình bình phương gradient theo công thức sau:

Trong đó:

* là trung bình bình phương gradient tại thời điểm t,
* là hệ số giảm xuống, thường được đặt giá trị mặc định là 0.9

Bước 4: Cập nhật trọng số θ bằng cách sử dụng tỷ lệ học (η) chia cho căn bậc hai của trung bình bình phương gradient đã tính:

Trong đó:

* là tỷ lệ học (learning rate)
* là một giá trị nhỏ (thường là 10-8) để tránh chia cho 0

Bước 5: Lặp các bước trên cho mỗi epoch hoặc mini-batch cho đến khi đạt được tiêu chí dừng hoặc số lượng epoch đã đủ.

RMSprop giữ trung bình bình phương gradient để điều chỉnh tỷ lệ học theo thời gian. Bằng cách làm điều này, nó giúp kiểm soát tỷ lệ học và giảm khả năng quá mức điều chỉnh trong quá trình tối ưu hóa mô hình.

1. Bài toán phù hợp

Phương pháp tối ưu hóa RMSprop thường phù hợp với nhiều loại dữ liệu và bài toán machine learning. Dưới đây là một số tình huống mà RMSprop có thể được ưa chuộng:

* Dữ Liệu Nhiễu: Khi dữ liệu có nhiễu hoặc biến động mạnh, RMSprop có thể giúp ổn định quá trình tối ưu hóa bằng cách điều chỉ tỷ lệ học dựa trên độ lớn của gradient.
* Mô Hình Học Sâu (Deep Learning): RMSprop thường được sử dụng trong deep learning, đặc biệt là khi đối mặt với các mô hình sâu và phức tạp có nhiều tham số.
* Bài toán Phân loại và Hồi quy: RMSprop thích hợp cho các bài toán phân loại và hồi quy, đặc biệt là khi có sự biến động trong dữ liệu đầu vào.
* Tập Dữ Liệu Lớn: Khi làm việc với tập dữ liệu lớn, RMSprop có thể giúp ổn định quá trình tối ưu hóa và giảm khả năng quá mức điều chỉnh.
* Dữ Liệu Đa Dạng: Khi dữ liệu có đặc trưng đa dạng, RMSprop có thể giúp kiểm soát tỷ lệ học cho từng tham số mô hình dựa trên độ lớn của gradient.
* Quá Trình Học Online: RMSprop có thể phù hợp trong quá trình học online, nơi mô hình cần được cập nhật khi có dữ liệu mới được thêm vào.
* Bài toán với Gradient Biến Động Nhanh Chóng: Khi gradient biến động nhanh chóng, RMSprop có thể giúp kiểm soát tỷ lệ học và tránh overshooting.

Tuy nhiên, không có một quy tắc cứng nhắc về việc lựa chọn optimizer, và thường cần thử nghiệm và so sánh với các phương pháp khác để xác định optimizer phù hợp nhất cho bài toán và dữ liệu cụ thể.

CHƯƠNG II: TÌM HIỂU VỀ CONTINUAL LEARNING VÀ TEST PRODUCTION

1. Continual Learning

Continual Learning (CL) trong machine learning là một lĩnh vực nghiên cứu và phát triển mô hình máy học có khả năng liên tục học từ dữ liệu mới mà không quên đi kiến thức đã học từ dữ liệu cũ, trong đó dữ liệu trong các lần học cũ không còn nữa trong quá trình học những cái mới.

Trong machine learning, các mô hình thường được huấn luyện trên một tập dữ liệu cố định và không thể học thêm sau khi huấn luyện ban đầu. Tuy nhiên, trong thực tế, dữ liệu có thể xuất hiện dưới dạng luồng không ngừng, và mô hình cần thích nghi và học từ các tác vụ mới liên tục mà không mất đi kiến thức đã học từ các tác vụ trước.

1. Điểm khác biệt của Continual Learning so với các mô hình Machine Learning truyền thống

So với các mô hình machine learning truyền thống thì Continual Learning có các điểm khác biệt sau:

* Đối tượng học: Trong các mô hình machine learning truyền thống, mô hình thường được huấn luyện trên một tập dữ liệu cố định và không có khả năng học thêm thông tin mới sau khi huấn luyện ban đầu. Trong khi đó, continual learning tập trung vào khả năng học liên tục và độc lập với dữ liệu, cho phép mô hình tiếp tục học từ các tác vụ mới mà không quên đi kiến thức đã học từ các tác vụ trước.
* Quên đi và xáo trộn kiến thức: Continual learning đối mặt với hai thách thức chính là quên đi (catastrophic forgetting) và sự xáo trộn kiến thức (knowledge transfer). Trong các mô hình truyền thống, không có cơ chế tự động để xử lý hai thách thức này. Trong khi đó, continual learning tự động giải quyết hai vấn đề này bằng cách duy trì kiến thức quan trọng từ các tác vụ trước và áp dụng kiến thức đó cho các tác vụ mới.
* Dữ liệu liên tục: Trong các mô hình truyền thống, dữ liệu thường được coi là tĩnh và không thay đổi sau khi huấn luyện. Trong khi đó, continual learning giả định rằng dữ liệu xuất hiện dưới dạng luồng không ngừng và mô hình cần thích nghi và học từ các tác vụ mới liên tục. Điều này đòi hỏi continual learning phải có khả năng xử lý dữ liệu động và học từ các tác vụ mới khi chúng xuất hiện.
* Phương pháp học: Các mô hình machine learning truyền thống thường sử dụng phương pháp học giám sát và đòi hỏi một lượng lớn dữ liệu được gán nhãn để huấn luyện. Trong khi đó, continual learning có thể sử dụng các phương pháp học không giám sát, học bán giám sát hoặc học tăng cường để học từ các tác vụ mới. Điều này mở ra khả năng sử dụng các nguồn dữ liệu không gán nhãn hoặc tự động thu thập dữ liệu mới trong quá trình học liên tục.

Các điểm khác biệt này chỉ ra sự chuyển đổi từ mô hình machine learning truyền thống sang mô hình có khả năng liên tục học trong CL, giúp mô hình thích ứng với môi trường động và không ngừng cập nhật kiến thức, làm cho continual learning trở thành một lĩnh vực đặc biệt trong machine learning, tập trung vào việc xây dựng mô hình có khả năng học liên tục và linh hoạt, giúp chúng tự động học và mở rộng kiến thức từ các tác vụ mới trong thực tế.

Tuy nhiên continual learning cũng có những thách thức giống như các mô hình thông thường, và ta cũng có những cách để giải quyết được những thách thức đó.

1. Những thách thức của Continual Learning và giải pháp
2. Quên mất toàn bộ (Catastrophic forgetting)

Catastrophic forgetting (quên mất toàn bộ) là hiện tượng mô hình machine learning quên đi hoàn toàn kiến thức đã học từ các tác vụ trước khi học tác vụ mới. Khi một mô hình được huấn luyện trên một tác vụ cụ thể, các tham số của mô hình được điều chỉnh để phù hợp với dữ liệu của tác vụ đó. Tuy nhiên, khi mô hình chuyển sang học tác vụ mới, quá trình huấn luyện trên tác vụ mới có thể làm thay đổi các tham số đã được điều chỉnh trước đó, dẫn đến việc quên mất kiến thức đã học từ tác vụ trước.

Catastrophic forgetting có thể xảy ra vì các tham số của mô hình không được giữ nguyên trong quá trình học liên tục. Khi mô hình được huấn luyện trên tác vụ mới, các gradient của tác vụ mới thường ghi đè lên các gradient của tác vụ trước, làm thay đổi các tham số liên quan và làm mất đi các tri thức đã học từ các tác vụ trước. Điều này dẫn đến việc mô hình không thể truy cập và sử dụng kiến thức đã học trước đó khi cần thiết.

Catastrophic forgetting có thể gây ra những vấn đề nghiêm trọng trong continual learning. Nếu mô hình không thể giữ lại kiến thức quan trọng từ các tác vụ trước, hiệu suất của nó trên các tác vụ trước đó sẽ giảm đi đáng kể.

Để giải quyết vấn đề này, một số phương pháp đã được đề xuất:

* Regularization-based approaches: Các phương pháp này như Elastic Weight Consolidation (EWC) và Synaptic Intelligence giữ lại kiến thức quan trọng từ các tác vụ trước bằng cách giới hạn sự thay đổi của các tham số quan trọng trong quá trình học tác vụ mới. Các tham số quan trọng được xác định dựa trên tầng quan sát (observed layer) và đánh giá sự quan trọng của chúng đối với các tác vụ trước.
* Replay-based approaches: Các phương pháp này sử dụng các bộ nhớ để lưu trữ và tái sử dụng các mẫu dữ liệu từ các tác vụ trước. Bằng cách lựa chọn một số mẫu dữ liệu từ các tác vụ trước và sử dụng chúng trong quá trình huấn luyện tác vụ mới, mô hình có thể nhớ lại và bảo tồn kiến thức đã học trước đó.
* Dynamic architecture approaches: Các phương pháp này tập trung vào việc tạo ra các kiến trúc mô hình linh hoạt có khả năng thay đổi kích thước, số lượng tầng ẩn hoặc số lượng đơn vị. Bằng cách điều chỉnh kiến trúc mô hình theo cách tốt nhất cho mỗi tác vụ, mô hình có khả năng học tác vụ mới mà không gây quên mất kiến thức đã học từ các tác vụ trước.

Các phương pháp trên giúp giải quyết vấn đề catastrophic forgetting trong continual learning. Tuy nhiên, vẫn còn nhiều nghiên cứu và phát triển trong lĩnh vực này để tìm ra các phương pháp hiệu quả hơn và cải thiện hiệu suất của mô hình trong việc học liên tục.

1. Xáo trộn kiến thức (Knowledge transfer)

Xáo trộn kiến thức (Knowledge transfer) là quá trình chia sẻ và áp dụng kiến thức đã học từ các tác vụ trước để cải thiện hiệu suất học của mô hình trong các tác vụ mới. Khi một mô hình học liên tục, nó có thể học được nhiều kiến thức từ các tác vụ trước đó. Tuy nhiên, việc áp dụng trực tiếp kiến thức từ một tác vụ sang tác vụ khác có thể gặp khó khăn do sự khác biệt về cấu trúc dữ liệu hoặc yêu cầu khác nhau của các tác vụ.

Để giải quyết vấn đề này, có một số phương pháp và kỹ thuật sau:

* Replay-based approaches: Phương pháp này lưu trữ và tái sử dụng các mẫu dữ liệu từ các tác vụ trước. Bằng cách lựa chọn một số mẫu dữ liệu từ các tác vụ trước và sử dụng chúng trong quá trình huấn luyện tác vụ mới, mô hình có thể học từ các trường hợp đã học trước đó. Các mẫu dữ liệu này có thể được lưu trữ trong một bộ nhớ đệm (buffer) và được sử dụng trong quá trình huấn luyện tác vụ mới, cung cấp cho mô hình một tập dữ liệu đa dạng để học.
* Feature extraction: Một cách phổ biến để chia sẻ kiến thức là thông qua việc trích xuất đặc trưng. Các tác vụ trước đó có thể đã học được các đặc trưng cần thiết cho các tác vụ mới. Bằng cách sử dụng các tầng đầu tiên của mô hình đã học trước đó như một trình trung gian, ta có thể trích xuất các đặc trưng từ dữ liệu mới và sử dụng chúng để huấn luyện một tầng phân loại riêng cho tác vụ mới.
* Knowledge distillation: Phương pháp này nhằm truyền đạt kiến thức từ mô hình phức tạp (teacher model) đã học trước đó sang một mô hình đơn giản hơn (student model). Mô hình phức tạp có thể có khả năng đạt được hiệu suất cao trên các tác vụ trước. Bằng cách huấn luyện một mô hình đơn giản với mục tiêu là sao chép phản ứng của mô hình phức tạp, ta có thể truyền đạt kiến thức từ mô hình phức tạp sang mô hình đơn giản hơn.
* Meta-learning: Phương pháp meta-learning tập trung vào việc học cách học nhanh và hiệu quả từ các tác vụ mới. Thay vì chỉ học từ các tác vụ trước, mô hình meta-learning học cách tìm ra các mẫu, quy tắc hoặc cấu trúc chung giữa các tác vụ và áp dụng chúng để học nhanh và hiệu quả trên các tác vụ mới.

Các phương pháp và kỹ thuật trên giúp chia sẻ và áp dụng kiến thức đã học từ các tác vụ trước để cải thiện hiệu suất học của mô hình trong cáctác vụ mới. Xáo trộn kiến thức đóng vai trò quan trọng trong continual learning và là một lĩnh vực nghiên cứu đang phát triển để tối ưu hóa việc chia sẻ và áp dụng kiến thức giữa các tác vụ khác nhau.

1. Quản lý tài nguyên

Continual learning yêu cầu quản lý tài nguyên huấn luyện mô hình, bao gồm bộ nhớ và khả năng tính toán. Khi số lượng tác vụ và kích thước dữ liệu tăng lên, việc lưu trữ và xử lý tất cả các dữ liệu và mô hình trở nên phức tạp và tốn kém.

**Giải pháp:** Có thể sử dụng các phương pháp như dynamic architecture approaches để tạo ra các kiến trúc mô hình linh hoạt có khả năng thay đổi kích thước, số lượng tầng ẩn hoặc số lượng đơn vị, từ đó tối ưu hóa việc sử dụng tài nguyên. Ngoài ra, các kỹ thuật nén mô hình có thể được áp dụng để giảm kích thước của mô hình và tăng khả năng quản lý tài nguyên.

1. Test Production
2. Test Production là gì?

Trong quá trình phát triển và triển khai các mô hình machine learning, "test production" là một pha trong quy trình kiểm tra và đánh giá hiệu suất của mô hình trước khi triển khai nó vào môi trường sản xuất (production environment).

Trong giai đoạn này, sau khi mô hình đã được huấn luyện và tinh chỉnh trên tập dữ liệu huấn luyện và tập dữ liệu xác thực, nó được đánh giá trên một tập dữ liệu kiểm tra độc lập. Mục tiêu là đánh giá hiệu suất của mô hình trên tập dữ liệu mà nó chưa từng thấy trước đây, để đảm bảo rằng nó có khả năng tổng quát hóa và dự đoán tốt trên các dữ liệu mới.

1. Các bước của quy trình test production
2. **Chuẩn bị dữ liệu test**

Cần chuẩn bị một tập dữ liệu kiểm tra độc lập, không được sử dụng trong quá trình huấn luyện hoặc xác thực. Dữ liệu kiểm tra này nên đại diện cho các điều kiện và tình huống mà mô hình sẽ gặp phải trong môi trường sản xuất.

Tập dữ liệu ban đầu được chia thành ba phần chính: tập dữ liệu huấn luyện, tập dữ liệu xác thực và tập dữ liệu kiểm tra. Trong đó, tập dữ liệu huấn luyện được sử dụng để huấn luyện mô hình, tập dữ liệu xác thực được sử dụng để điều chỉnh các siêu tham số và tập dữ liệu kiểm tra được sử dụng để đánh giá hiệu suất cuối cùng của mô hình.

1. **Chuẩn bị môi trường test**

Cần tạo ra một môi trường tương tự như môi trường thực tế mà mô hình sẽ hoạt động trong đó. Điều này bao gồm cài đặt các thư viện, phần mềm và cấu hình cần thiết để chạy mô hình và triển khai nó.

1. **Chạy mô hình trên dữ liệu kiểm tra**

Mô hình được chạy trên tập dữ liệu kiểm tra và tạo ra các dự đoán. Các đầu ra của mô hình được ghi lại để phân tích và đánh giá hiệu suất sau này.

1. **Đánh giá hiệu suất**

Sau khi mô hình đã được huấn luyện hoàn tất và điều chỉnh tối ưu, nó được đánh giá trên tập dữ liệu kiểm tra độc lập. Mục tiêu là đo lường hiệu suất của mô hình trên dữ liệu mà nó chưa từng thấy trước đó và đánh giá khả năng tổng quát hóa của nó. Kết quả này sẽ cho thấy khả năng dự đoán và độ chính xác của mô hình trên dữ liệu mới.

1. **Phân tích kết quả**

Kết quả từ quá trình testing bao gồm các phép đo hiệu suất như độ chính xác, độ phân loại, độ lỗi, đánh giá F1,... Dựa trên các kết quả này, chúng ta có thể đánh giá và so sánh hiệu suất của mô hình và thực hiện các điều chỉnh cần thiết (overfitting,…).

1. **Cải thiện mô hình**

Dựa trên kết quả của quá trình testing, nhà phát triển có thể thấy điểm yếu hoặc vấn đề trong mô hình và thực hiện các điều chỉnh để cải thiện hiệu suất. Điều này có thể bao gồm việc thay đổi tham số, chọn lại thuật toán, tăng cường dữ liệu,…

1. Mục đích của Test Production

Quá trình test production đảm bảo rằng mô hình đã được huấn luyện và triển khai hoạt động tốt trong môi trường sản xuất. Test production có vai trò quan trọng để đảm bảo rằng mô hình có khả năng tổng quát hóa và dự đoán tốt trên dữ liệu mới, không chỉ trên dữ liệu huấn luyện và xác thực.

Dưới đây là những mục đích cụ thể của quy trình:

* Đánh giá hiệu suất: Test production giúp đánh giá hiệu suất của mô hình trên tập dữ liệu kiểm tra độc lập. Mục tiêu là đo lường độ chính xác, độ phân loại, độ lỗi hoặc các phép đo khác để đánh giá khả năng dự đoán của mô hình trên dữ liệu mới.
* Kiểm tra khả năng tổng quát hóa: Test production giúp đảm bảo rằng mô hình có khả năng tổng quát hóa, đảm bảo rằng mô hình không chỉ hoạt động tốt trên tập dữ liệu huấn luyện và xác thực mà còn trên các dữ liệu mới, không từng được mô hình gặp phải trước đó.
* Xác nhận tính ổn định: Test production cũng giúp xác nhận tính ổn định của mô hình trong môi trường sản xuất. Môi trường sản xuất có thể có các biến đổi và điều kiện khác nhau so với môi trường huấn luyện, và test production giúp đảm bảo rằng mô hình vẫn hoạt động ổn định và đáng tin cậy trong những điều kiện thực tế.
* Tối ưu hóa mô hình: Kết quả từ test production cung cấp thông tin quan trọng để tối ưu hoá mô hình. Nếu mô hình không đạt yêu cầu hoặc không có hiệu suất đủ tốt trong môi trường sản xuất, nhà phát triển có thể thực hiện các điều chỉnh và cải thiện để nâng cao hiệu suất và khả năng tổng quát hóa.
* Đảm bảo chất lượng sản phẩm: Test production đóng vai trò quan trọng trong việc đảm bảo chất lượng sản phẩm hoặc dịch vụ mà mô hình machine learning đang hỗ trợ. Bằng cách đánh giá mô hình trong môi trường sản xuất, ta có thể xác định xem mô hình có đáp ứng được yêu cầu và tiêu chuẩn chất lượng hay không.

Tóm lại, mục đích của test production là đảm bảo rằng mô hình machine learning có hiệu suất tốt, khả năng tổng quát hóa và đáng tin cậy trong môi trường sản xuất. Nó giúp xác định và nâng cao chất lượng sản phẩm và đảm bảo rằng mô hình hoạt động tốt trong thực tế.

# TÀI LIỆU THAM KHẢO

1. Tài liệu về các optimizer được tra cứu trên Google và Wikipedia
2. Video giảng về Cotinual Learning và Test Production trên Youtube
3. Slide bài giảng nhập môn học máy của trường Đại học Tôn Đức Thắng