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

ĐỒ ÁN CUỐI KÌ

MÔN NHẬP MÔN HỌC MÁY

TRÌNH BÀY NGHIÊN CỨU, ĐÁNH GIÁ PHẦN LÝ THUYẾT

Người hướng dẫn: **GV LÊ ANH CƯỜNG**

Người thực hiện: **PHAN ĐĂNG KHÔI - 52100241**

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

ĐỒ ÁN CUỐI KÌ

MÔN NHẬP MÔN HỌC MÁY

TRÌNH BÀY NGHIÊN CỨU, ĐÁNH GIÁ PHẦN LÝ THUYẾT

Người hướng dẫn: **GV LÊ ANH CƯỜNG**
Người thực hiện: **PHAN ĐĂNG KHÔI - 52100241**
Lớp **: 21050301**
Khoá **: 25**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023

LỜI CẢM ƠN

Lời đầu tiên chúng em chân thành gửi lời cảm ơn đến **Trường đại học Tôn Đức Thắng** đã cho chúng em cơ hội được học và tiếp cận môn học này, giúp chúng em định hướng và có mục tiêu rõ ràng trong tương lai.

Chúng em xin chân thành gửi lời cảm ơn đến **thầy Lê Anh Cường** vì sự nhiệt tình của thầy trong việc giảng dạy, bên cạnh đó là những kiến thức và kinh nghiệm thầy truyền đạt giúp em hiểu biết hơn về các thuật toán và kiến thức nền tảng của môn học, tuy chỉ trong vài tháng ngắn ngủi được học tập và làm việc với thầy nhưng những kiến thức thầy truyền đạt là rất quý giá.

Trong quá trình làm bài báo cáo này, với kinh nghiệm và kiến thức còn ít ỏi nên không tránh được những sơ sót và hạn chế, em mong có thể nhận được sự góp ý của thầy để bài báo cáo được hoàn chỉnh hơn, hiệu quả hơn. Cuối cùng chúng em xin gửi lời chúc sức khoẻ đến thầy, chúc thầy luôn vui vẻ, hạnh phúc và thành công trong công cuộc lèo lái con đò đưa trò sang sông.

Tôi xin chân thành cảm ơn!

Trân trọng.

CAM KẾT
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 thầy 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 phát hiện có bất kỳ sự gian lận nào tôi 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 tôi gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày 18 tháng 12 năm 2023

Tác giả

(ký tên và ghi rõ họ tên)

Phan Đăng Khôi

MỤC LỤC

LỜI CẢM ƠN i
CAM KẾT ii
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 chung. 3
1.2 Một số optimizer được sử dụng phổ biến. 3
1.2.1 Gradient Descent (GD) 3
1.2.2 Stochastic Gradient Descent (SGD). 4
1.2.3 Stochastic Gradient Descent with Momentum 4
1.2.4 Mini-Batch Gradient Descent. 5
1.2.5 Adagrad 5
1.2.6 RMSProp 6
1.2.7 AdaDelta 6
1.2.8 Adam 6
1.3 So sánh ưu nhược điểm của các thuật toán 7
1.4 Cách thức hoạt động trong mô hình 9
1.5 Kết luận 10
CHƯƠNG 2 – TÌM HIỂU VỀ CONTINUAL LEARNING VÀ TEST PRODUCTION KHI XÂY DỰNG MỘT GIẢI PHÁP HỌC MÁY. 12
2.1 Tìm hiểu chung 12
2.1.1 Continual Learning 12
2.1.2 Test Production 13
2.2 Xây dựng bài toán với Continual Learning và Test Production 14

2.2.1 Xây dựng bài toán với Continual Learning 14
2.2.2 Xây dựng bài toán áp dụng Test Production 15
TÀI LIỆU THAM KHẢO 17
Tiếng Việt 17
Tiếng Anh 17

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

1.1 Tìm hiểu chung.

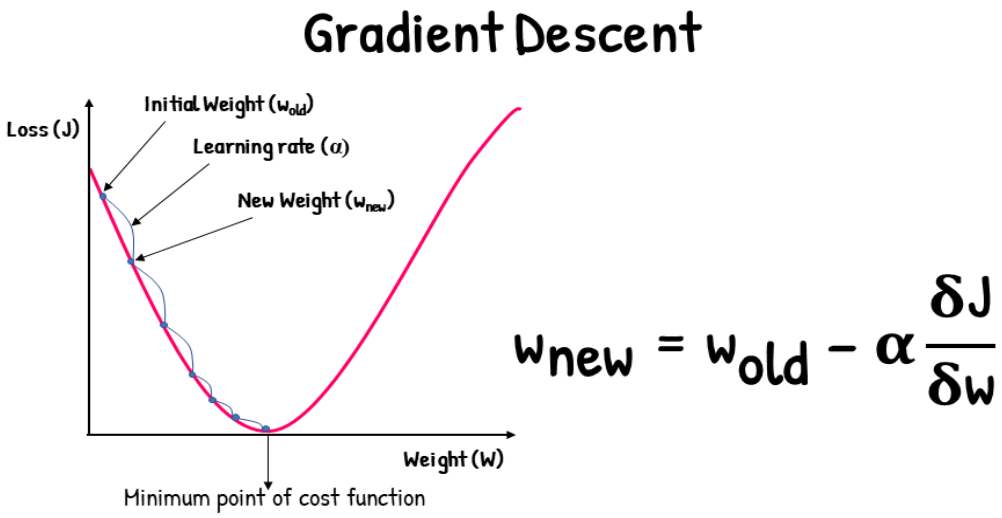
Các thuật toán tối ưu hóa là một phần quan trọng trong quá trình đào tạo các mô hình Machine Learning. Họ chịu trách nhiệm điều chỉnh các tham số của mô hình để giảm thiểu hàm mất mát, hàm này đo lường mức độ mô hình có thể đưa ra dự đoán trên một tập dữ liệu nhất định. Có sẵn các thuật toán tối ưu hóa khác nhau và việc chọn thuật toán nào có thể tác động đáng kể đến hiệu suất của mô hình.

Optimizers là các thuật toán được sử dụng để điều chỉnh các tham số của mô hình nhằm giảm thiểu hàm mất mát. Việc lựa chọn trình tối ưu hóa có thể ảnh hưởng lớn đến hiệu suất và tốc độ đào tạo mô hình.

1.2 Một số optimizer được sử dụng phổ biến.

1.2.1 Gradient Descent (GD)

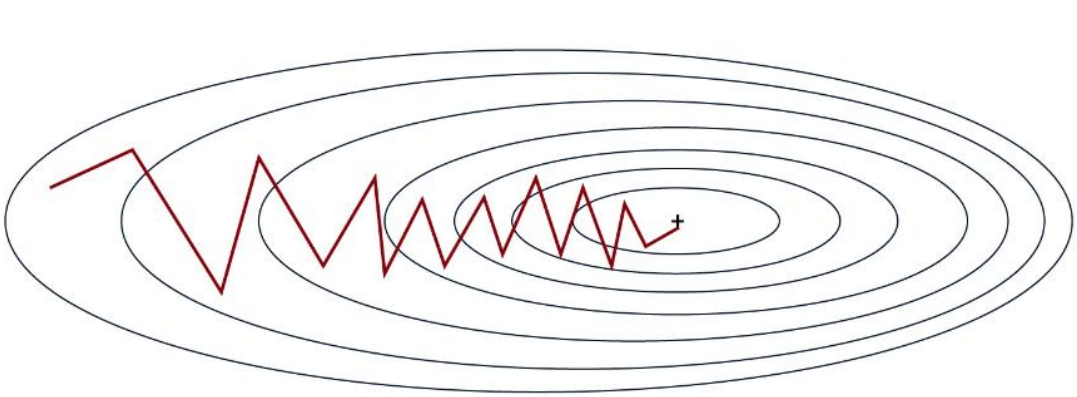
GD là một phương pháp tối ưu đơn giản và cơ bản nhất. Nó cập nhật các tham số theo hướng ngược của gradient của hàm mất mát. Chúng ta có thể viết dạng cơ bản của thuật toán như sau: $w_{new} = w_{old} - \alpha \frac{\delta J}{\delta w}$



1.2.2 Stochastic Gradient Descent (SGD).

Stochastic là 1 biến thể của Gradient Descent. Thay vì sau mỗi epoch chúng ta sẽ cập nhật trọng số (Weight) 1 lần thì trong mỗi epoch có N điểm dữ liệu chúng ta sẽ cập nhật trọng số N lần. Nhìn vào 1 mặt, SGD sẽ làm giảm đi tốc độ của 1 epoch. Tuy nhiên nhìn theo 1 hướng khác, SGD sẽ hội tụ rất nhanh chỉ sau vài epoch. Chúng ta có thể viết dạng cơ bản của thuật toán như sau: $\theta = \theta - \alpha \cdot \nabla_\theta L\left(\theta; x^{(i)}; y^{(i)}\right)$

$\boldsymbol{x^{(i)}; y^{(i)}}$ là một khối dữ liệu nhỏ.



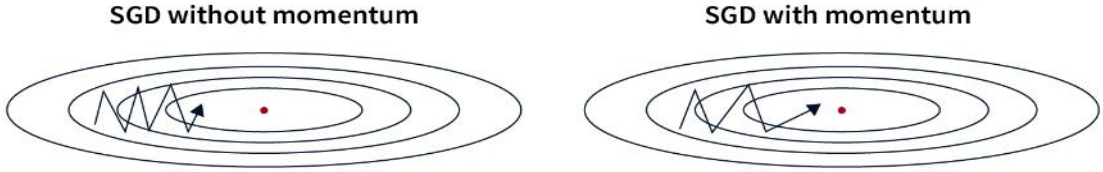
1.2.3 Stochastic Gradient Descent with Momentum

SGD with Momentum là một biến thể của SGD bổ sung thuật ngữ "Momentum" vào quy tắc cập nhật, giúp trình tối ưu hóa tiếp tục di chuyển theo cùng một hướng ngay cả khi độ dốc cục bộ nhỏ. Thuật ngữ "Momentum" thường được đặt ở giá trị trong khoảng từ 0 đến 1. Chúng ta có thể viết quy tắc cập nhật như sau:

$$v = \beta . v + (1 - \beta) . \nabla_\theta L\left(\theta; x^{(i)}; y^{(i)}\right)$$

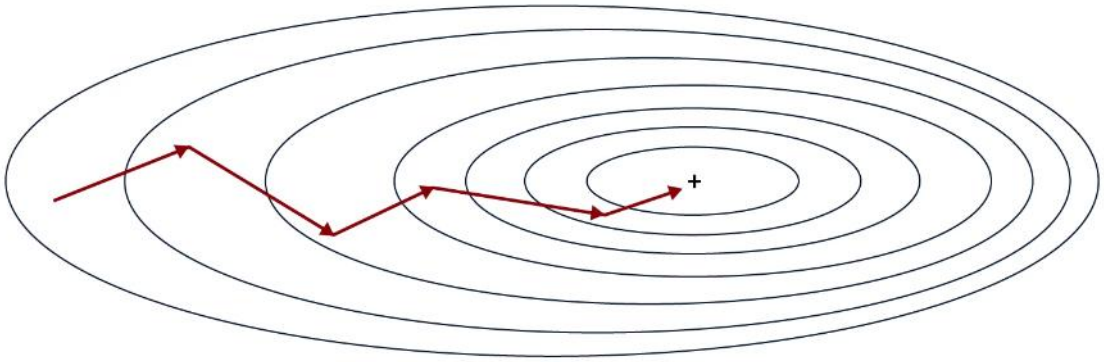
$$\theta = \theta - \alpha . v$$

α là vectơ động lượng và β là siêu tham số động lượng.



1.2.4 Mini-Batch Gradient Descent.

Mini-Batch Gradient Descent tương tự như SGD, nhưng thay vì sử dụng một mẫu duy nhất để tính toán độ dốc, nó sử dụng một "lô nhỏ" có kích thước cố định gồm các mẫu. Quy tắc cập nhật giống như đối với SGD, ngoại trừ độ dốc được tính trung bình trên lô nhỏ. Điều này có thể giảm nhiễu trong các bản cập nhật và cải thiện khả năng hội tụ.



1.2.5 Adagrad

Adagrad là một thuật toán tối ưu hóa sử dụng tốc độ học thích ứng cho mỗi tham số. Tốc độ học được cập nhật dựa trên thông tin độ dốc lịch sử để các tham số nhận được nhiều cập nhật có tốc độ học thấp hơn và các tham số nhận được ít cập nhật hơn có tốc độ học lớn hơn. Quy tắc cập nhật có thể được viết như sau:

$$g = \nabla_\theta L\left(\theta; x^{(i)}; y^{(i)}\right)$$

$$G = G + g \odot g$$

$$\theta = \theta - \frac{\alpha}{\sqrt{G + \varepsilon}} \odot g$$

G là một ma trận tích lũy các bình phương của độ dốc và ε là một hằng số nhỏ được thêm vào để tránh chia cho 0.

1.2.6 RMSProp

RMSProp là một thuật toán tối ưu hóa tương tự như Adagrad, nhưng nó sử dụng giá trị trung bình giảm dần theo cấp số nhân của bình phương các gradient thay vì tổng. Điều này giúp giảm sự suy giảm tốc độ học đơn điệu của Adagrad và cải thiện khả năng hội tụ. Chúng ta có thể viết quy tắc cập nhật như sau:

$$g = \nabla_\theta L\left(\theta; x^{(i)}; y^{(i)}\right)$$

$$G = \beta . G + (1 - \beta) . g \odot g$$

$$\theta = \theta - \frac{\alpha}{\sqrt{G + \varepsilon}} \odot g$$

G là một ma trận tích lũy các bình phương của gradient, ε là một hằng số nhỏ được thêm vào để tránh chia cho 0 và β là một siêu tham số tốc độ phân rã.

1.2.7 AdaDelta

AdaDelta là một thuật toán tối ưu hóa tương tự như RMSProp nhưng không yêu cầu tốc độ học siêu tham số. Thay vào đó, nó sử dụng giá trị trung bình giảm dần theo cấp số nhân của độ dốc và bình phương của độ dốc để xác định thang đo được cập nhật. Chúng ta có thể viết quy tắc cập nhật như sau:

$$g = \nabla_\theta L\left(\theta; x^{(i)}; y^{(i)}\right)$$

$$G = \beta . G + (1 - \beta) . g \odot g$$

$$\Delta\theta = -\frac{\sqrt{S + \varepsilon}}{\sqrt{G + \varepsilon}} \odot g$$

$$S = \beta . S + (1 - \beta) . \Delta\theta \odot \Delta\theta$$

$$\theta = \theta + \Delta\theta$$

G và S là các ma trận tích lũy độ dốc và bình phương của các bản cập nhật tương ứng và ε là một hằng số nhỏ được thêm vào để tránh phép chia cho 0.

1.2.8 Adam

Adam (viết tắt của "adaptive moment estimation") là một thuật toán tối ưu hóa kết hợp các ý tưởng của SGD with Momentum và RMSProp. Nó sử dụng giá trị trung

bình giảm dần theo cấp số nhân của độ dốc và bình phương của độ dốc để xác định thang đo được cập nhật, tương tự như RMSProp. Nó cũng sử dụng thuật ngữ Momentum để giúp trình tối ưu hóa di chuyển hiệu quả hơn thông qua hàm mất mát. Quy tắc cập nhật có thể được viết như sau:

$$g = \nabla_\theta L\left(\theta; x^{(i)}; y^{(i)}\right)$$

$$m = \beta_1 . m + (1 - \beta_1) . g$$

$$v = \beta_2 . v + (1 - \beta_2) . g \odot g$$

$$\hat{m} = \frac{m}{1 - \beta_1^t}$$

$$\hat{v} = \frac{v}{1 - \beta_2^t}$$

$$\theta = \theta - \frac{\alpha}{\sqrt{\hat{v} + \varepsilon}} \odot \hat{m}$$

m và v lần lượt là các vectơ động lượng và vận tốc, $\beta_1$và β_2 là tốc độ phân rã của động lượng và vận tốc.

1.3 So sánh ưu nhược điểm của các thuật toán

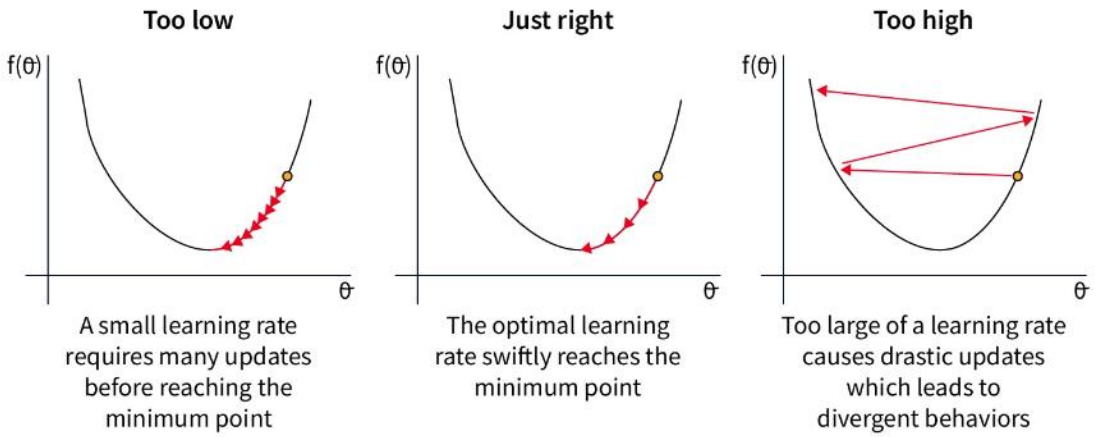
Thuật toán	Ưu điểm	Nhược điểm
Gradient Descent	Đơn giản để thực hiện. Có thể hoạt động tốt với tốc độ học tập được điều chỉnh tốt.	Có thể hội tụ chậm, đặc biệt đối với các mô hình phức tạp hoặc tập dữ liệu lớn. Nhạy cảm với việc lựa chọn tốc độ học tập.
Adagrad	Có thể hoạt động tốt với dữ liệu thưa thớt.	Có thể hội tụ quá chậm đối với một số vấn đề.

	Tự động điều chỉnh tốc độ học tập dựa trên các cập nhật tham số.	Có thể ngừng học hoàn toàn nếu tốc độ học trở nên quá nhỏ.
Stochastic Gradient Descent	Có thể nhanh hơn Gradient Descent, đặc biệt là đối với các tập dữ liệu lớn. Có thể thoát khỏi cực tiểu địa phương dễ dàng hơn.	Kém ổn định hơn. Yêu cầu điều chỉnh siêu tham số nhiều hơn để đạt hiệu suất tốt hơn.
Stochastic Gradient Descent with Momentum	Giúp di chuyển hiệu quả hơn qua các vùng "phẳng" của hàm mất mát. Giúp giảm dao động và cải thiện sự hội tụ.	Có thể vượt qua các giải pháp tốt và giải quyết các giải pháp dưới mức tối ưu nếu đà quá cao. Yêu cầu điều chỉnh siêu tham số động lượng.
Mini – Bathch Gradient Descent	Có thể nhanh hơn Gradient Descent, đặc biệt là đối với các tập dữ liệu lớn. Có thể thoát khỏi cực tiểu địa phương dễ dàng hơn. Có thể giảm nhiễu trong các bản cập nhật, dẫn đến sự hội tụ ổn định hơn.	Có thể nhạy cảm với việc lựa chọn kích thước lô nhỏ.
RMSProp	Có thể hoạt động tốt với dữ liệu thưa thớt.	Có thể hội tụ quá chậm đối với một số vấn đề. Yêu cầu điều chỉnh siêu tham số tốc độ phân rã.

	Tự động điều chỉnh tốc độ học tập dựa trên các cập nhật tham số. Có thể hội tụ nhanh hơn Adagrad.	
AdaDelta	Có thể hoạt động tốt với dữ liệu thưa thớt. Tự động điều chỉnh tốc độ học tập dựa trên các cập nhật tham số.	Có thể hội tụ quá chậm đối với một số vấn đề. Có thể ngừng học hoàn toàn nếu tốc độ học trở nên quá nhỏ.
Adam	Có thể hội tụ nhanh hơn các thuật toán tối ưu hóa khác. Hoạt động tốt với dữ liệu lớn.	Yêu cầu điều chỉnh nhiều siêu tham số hơn các thuật toán khác.

1.4 Cách thức hoạt động trong mô hình

Trình tối ưu hóa trong học máy điều chỉnh các tham số của mô hình để giảm thiểu hàm mất mát. Hàm mất mát đo lường mức độ mô hình có thể đưa ra dự đoán trên một tập dữ liệu nhất định và mục tiêu của việc huấn luyện mô hình là tìm ra tập hợp các tham số mô hình mang lại tổn thất thấp nhất có thể.



Trình tối ưu hóa sử dụng thuật toán tối ưu hóa để tìm kiếm các tham số giảm thiểu hàm mất mát. Thuật toán tối ưu hóa sử dụng độ dốc của hàm mất mát đối với các tham số mô hình để xác định hướng mà chúng ta nên điều chỉnh các tham số.

Độ dốc được tính toán bằng cách sử dụng lan truyền ngược, bao gồm việc áp dụng quy tắc chuỗi để tính toán độ dốc của hàm mất mát cho từng tham số mô hình.

Thuật toán tối ưu hóa sau đó điều chỉnh các tham số mô hình để giảm thiểu hàm mất mát. Quá trình này được lặp lại cho đến khi hàm mất mát đạt đến mức tối thiểu hoặc trình tối ưu hóa đạt đến số lần lặp tối đa được phép.

1.5 Kết luận

Trình tối ưu hóa trong học máy hay học sâu là rất cần thiết vì chúng điều chỉnh các tham số của mô hình để giảm thiểu hàm mất mát.

Nói chung, việc lựa chọn sử dụng thuật toán tối ưu hóa nào sẽ phụ thuộc vào đặc điểm cụ thể của vấn đề, chẳng hạn như kích thước của tập dữ liệu và độ phức tạp của mô hình.

Điều quan trọng là phải xem xét cẩn thận ưu và nhược điểm của từng thuật toán và điều chỉnh mọi siêu tham số có liên quan để đạt được hiệu suất tốt nhất có thể.

Nhìn chung, hiểu được vai trò của trình tối ưu hóa trong học sâu và các thuật toán khác nhau có sẵn là điều cần thiết đối với bất kỳ ai muốn xây dựng và đào tạo các mô hình học máy hiệu quả.

CHƯƠNG 2 – TÌM HIỂU VỀ CONTINUAL LEARNING VÀ TEST PRODUCTION KHI XÂY DỰNG MỘT GIẢI PHÁP HỌC MÁY.

2.1 Tìm hiểu chung

2.1.1 Continual Learning

Continual Learning (hoặc còn gọi là Lifelong Learning hoặc Incremental Learning) là khái niệm trong lĩnh vực học máy, nơi mô hình học máy được huấn luyện và cải tiến liên tục khi đối mặt với dữ liệu mới hoặc các nhiệm vụ mới, mà không quên đi kiến thức đã học trước đó. Mục tiêu của Continual Learning là xây dựng các mô hình học máy có khả năng tự động học và tích lũy kiến thức, từ đó đạt được khả năng học tập linh hoạt và giải quyết nhiều nhiệm vụ khác nhau.

Một số phương pháp và kỹ thuật phổ biến được sử dụng trong Continual Learning bao gồm:

- Replay-based Methods: Phương pháp này sử dụng lại dữ liệu huấn luyện từ các nhiệm vụ trước đó để giúp mô hình học máy nhớ lại kiến thức đã học. Dữ liệu từ các nhiệm vụ trước đó được lưu trữ trong bộ nhớ và kết hợp với dữ liệu mới để huấn luyện mô hình.
- Regularization-based Methods: Phương pháp này sử dụng các kỹ thuật regularization như Elastic Weight Consolidation (EWC) hoặc Synaptic Intelligence để bảo vệ kiến thức đã học trong quá trình huấn luyện mới. Các tham số quan trọng được ưu tiên bảo vệ để tránh quên đi kiến thức cũ.
- Dynamic Architecture Methods: Phương pháp này tập trung vào việc điều chỉnh kiến trúc mô hình để phù hợp với các nhiệm vụ mới. Các kiến trúc mạng thần kinh có thể mở rộng hoặc thu gọn để tăng khả năng học và phân loại các nhiệm vụ mới.

- Generative Replay Methods: Phương pháp này sử dụng mô hình sinh dữ liệu (generative model) để tạo ra dữ liệu giả mạo từ các nhiệm vụ trước đó. Dữ liệu giả tạo này được sử dụng để huấn luyện mô hình trên các nhiệm vụ mới, giúp giữ lại kiến thức đã học.

2.1.2 Test Production

Test Production là một khái niệm liên quan đến quá trình triển khai và áp dụng giải pháp học máy vào một bài toán cụ thể. Khi xây dựng một giải pháp học máy, quá trình Test Production bao gồm các bước sau:

- Chuẩn bị dữ liệu kiểm tra: Dữ liệu kiểm tra được chuẩn bị để đánh giá hiệu suất của giải pháp học máy trên bài toán cụ thể. Dữ liệu kiểm tra thường được chia thành tập dữ liệu kiểm tra và tập dữ liệu đánh giá.
- Triển khai mô hình: Mô hình học máy được triển khai để sử dụng trong môi trường thực tế. Quá trình triển khai có thể bao gồm việc triển khai mô hình trên một hệ thống phân tán, tích hợp mô hình vào các ứng dụng hoặc giao diện người dùng.
- Đánh giá hiệu suất: Mô hình được đánh giá hiệu suất trên tập dữ liệu kiểm tra hoặc tập dữ liệu đánh giá. Các phép đo đánh giá như độ chính xác (accuracy), độ nhạy (recall), độ chính xác dương tính (precision), và F1-score thường được sử dụng để đánh giá hiệu suất của mô hình.
- Tinh chỉnh và cải tiến: Dựa trên kết quả đánh giá, có thể thực hiện quá trình tinh chỉnh và cải tiến mô hình để nâng cao hiệu suất trên dữ liệu kiểm tra. Điều này có thể bao gồm việc điều chỉnh siêu tham số, thay đổi kiến trúc mô hình hoặc thực hiện các biện pháp khác để cải thiện kết quả.

Quá trình Test Production đóng vai trò quan trọng trong xác định tính khả thi và hiệu suất của giải pháp học máy trên bài toán cụ thể. Nó giúp đảm bảo rằng mô hình được triển khai hoạt động đúng và mang lại giá trị thực tế trong môi trường ứng dụng.

2.2 Xây dựng bài toán với Continual Learning và Test Production

2.2.1 Xây dựng bài toán với Continual Learning

Giả sử ta muốn xây dựng một giải pháp học máy để phân loại hình ảnh của các loài động vật, và bạn muốn áp dụng Continual Learning để mô hình có thể học và phân loại một loài động vật mới khi có dữ liệu mới được thêm vào.

Dưới đây là một cách thức để xây dựng bài toán này sử dụng Continual Learning:

- Chuẩn bị dữ liệu: Thu thập dữ liệu hình ảnh của các loài động vật khác nhau, ví dụ như hình ảnh của mèo, chó, chim và cá. Chia dữ liệu thành các tập dữ liệu huấn luyện ban đầu và tập dữ liệu kiểm tra.
- Huấn luyện mô hình ban đầu: Sử dụng một mô hình học máy, chẳng hạn một mạng neural convolutional (CNN), để huấn luyện trên tập dữ liệu huấn luyện ban đầu để phân loại các loài động vật. Đánh giá hiệu suất của mô hình trên tập dữ liệu kiểm tra.
- Mở rộng mô hình: Khi có một loài động vật mới được thêm vào, tiếp tục huấn luyện mô hình đã huấn luyện ban đầu bằng cách sử dụng tập dữ liệu mới này. Tuy nhiên, thay vì huấn luyện lại toàn bộ mô hình, tập trung huấn luyện các lớp cuối cùng (fully connected layers) để phân loại loài động vật mới.
- Replay-based Methods: Để giúp mô hình không quên đi kiến thức đã học, sử dụng phương pháp replay-based methods. Khi huấn luyện với dữ liệu mới của loài động vật mới, sử dụng một phần dữ liệu từ các loài động vật đã được huấn luyện trước đó để làm "replay data". Điều này giúp mô hình nhớ lại kiến thức đã học và giảm quên đi thông tin quan trọng.
- Đánh giá hiệu suất: Sau khi huấn luyện mô hình trên dữ liệu mới, đánh giá hiệu suất của mô hình trên tập dữ liệu kiểm tra mới bao gồm các loài động vật đã được huấn luyện trước đó và loài động vật mới.

- Lặp lại quá trình: Lặp lại quá trình mở rộng mô hình khi có loài động vật mới được thêm vào. Tiếp tục huấn luyện mô hình bằng cách sử dụng replay-based methods và đánh giá hiệu suất của mô hình sau mỗi lần mở rộng.

Qua các bước trên, mô hình sẽ có khả năng học và phân loại các loài động vật mới khi có dữ liệu mới được thêm vào, giúp mô hình tiếp tục nâng cao hiệu suất và tích lũy kiến thức theo thời gian.

2.2.2 Xây dựng bài toán áp dụng Test Production

Hãy xem xét một bài toán phân loại email là thư rác (spam) hoặc không phải thư rác (ham). Giả sử ta muốn xây dựng một giải pháp học máy để phân loại email và sau đó áp dụng Test Production để đánh giá hiệu suất của giải pháp trên dữ liệu kiểm tra.

Dưới đây là cách xây dựng bài toán này sử dụng Test Production:

- Chuẩn bị dữ liệu: Thu thập và chuẩn bị dữ liệu email, bao gồm các ví dụ của email là spam và email không phải là spam. Phân chia dữ liệu thành tập dữ liệu huấn luyện và tập dữ liệu kiểm tra.
- Tiền xử lý dữ liệu: Tiền xử lý dữ liệu email bằng cách loại bỏ các ký tự đặc biệt, chuyển đổi văn bản thành các vectơ đặc trưng (như TF-IDF hay Bag-of-Words), và chuẩn hóa dữ liệu nếu cần thiết.
- Xây dựng mô hình: Sử dụng một mô hình học máy, chẳng hạn như mạng neural hồi quy (RNN) hoặc Support Vector Machine (SVM), để huấn luyện trên tập dữ liệu huấn luyện. Đánh giá hiệu suất của mô hình trên tập dữ liệu kiểm tra ban đầu.
- Áp dụng Test Production: Đánh giá hiệu suất của mô hình trên tập dữ liệu kiểm tra. Sử dụng các phép đo đánh giá như độ chính xác (accuracy), độ nhạy (recall), độ chính xác dương tính (precision) và F1-score để đánh giá hiệu suất của mô hình trong việc phân loại email.

- Tinh chỉnh và cải tiến: Dựa trên kết quả đánh giá, thực hiện các biện pháp tinh chỉnh và cải tiến trên mô hình để nâng cao hiệu suất trên dữ liệu kiểm tra. Có thể thử nghiệm các phương pháp như tăng cường dữ liệu, thay đổi siêu tham số mô hình hoặc thử các mô hình khác nhau để tìm ra giải pháp tốt nhất.
- Đánh giá cuối cùng: Sau khi tinh chỉnh mô hình, đánh giá lại hiệu suất của mô hình trên tập dữ liệu kiểm tra cuối cùng. Điều này giúp đảm bảo rằng mô hình đã được cải thiện và sẵn sàng áp dụng vào thực tế.

Qua các bước trên, ta đã xây dựng một giải pháp học máy để phân loại email và áp dụng Test Production để đánh giá hiệu suất của giải pháp trên dữ liệu kiểm tra. Quá trình Test Production giúp đảm bảo rằng giải pháp học máy đã được thử nghiệm và đánh giá kỹ lưỡng trước khi triển khai trong một môi trường thực tế.

TÀI LIỆU THAM KHẢO

Tiếng Việt

1. https://machinelearningcoban.com/2017/01/16/gradientdescent2/
2. https://viblo.asia/p/optimizer-hieu-sau-ve-cac-thuat-toan-toi-uu-gdsgdadam-Qbq5QQ9E5D8
3. https://jos.husc.edu.vn/backup/upload/vol_18/no_1/668_fulltext_4.%C4%90TVT%20-%20Phuoc%20-%20Vuong%20Quang%20Phuoc.pdf

Tiếng Anh

1. https://medium.com/mlearning-ai/optimizers-in-deep-learning-7bf81fed78a0
2. https://www.geeksforgeeks.org/

PHỤ LỤC

Phần này bao gồm những nội dung cần thiết nhằm minh họa hoặc hỗ trợ cho nội dung luận văn như số liệu, biểu mẫu, tranh ảnh. . . . nếu sử dụng những câu trả lời cho một *bảng câu hỏi thì bảng câu hỏi mẫu này phải được đưa vào phần Phụ lục ở dạng nguyên bản* đã dùng để điều tra, thăm dò ý kiến; **không được tóm tắt hoặc sửa đổi**. Các tính toán mẫu trình bày tóm tắt trong các biểu mẫu cũng cần nêu trong Phụ lục của luận văn. Phụ lục không được dày hơn phần chính của luận văn