

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



ĐỒ ÁN CUỐI KỲ MÔN NHẬP MÔN HỌC MÁY

Người hướng dẫn: **LÊ VĂN CƯỜNG**

Người thực hiện: **NGÔ CẨM TÚ - 52100942**

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 KỲ MÔN NHẬP MÔN HỌC MÁY

Người hướng dẫn: **LÊ VĂN CƯỜNG**

Người thực hiện: **NGÔ CẨM TÚ**

Lớp : **21050301**

Khoá : **25**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023

LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành đến thầy Lê Văn Cường giảng viên bộ môn “Lê Văn Cường” trong Khoa Công nghệ Thông tin. Trong quá trình học tập và tìm hiểu bộ môn em đã nhận được sự giảng dạy tận tâm và vô cùng nhiệt huyết của thầy, nhờ đó đã trang bị cho em những kiến thức, kỹ năng cần thiết để hoàn thành bài giữa kỳ này. Thông qua bài báo cáo, em xin trình bày những gì mà mình đã tìm hiểu được trong quá trình học tập tại trường đại học Tôn Đức Thắng đến các thầy cô giảng viên và các thầy cô trong khoa Khoa Công nghệ Thông tin.

Trong quá trình nghiên cứu và làm bài báo cáo, do kiến thức chuyên ngành còn hạn chế nên em vẫn còn nhiều thiếu sót khi tìm hiểu, đánh giá và trình bày về bài báo cáo. Em rất được mong nhận những ý kiến nhận xét, ý kiến đóng góp và phê bình từ các thầy/cô giảng viên bộ môn trong khoa để bài báo cáo của em được đầy đủ và hoàn chỉnh hơn.

Em xin chân thành cảm ơn.

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 Lê Văn 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 **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 21 tháng 11 năm 2023

Tác giả

(ký tên và ghi rõ họ tên)

Tú

Ngô Cẩm Tú

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)

MỤC LỤC

NỘI DUNG BÁO CÁO	5
1. Tìm hiểu, so sánh các phương pháp Optimizer trong huấn luyện mô hình học máy	5
a. Phương pháp tối ưu (Optimizer) trong học máy là gì?	5
b. Các thuật toán tối ưu trong học máy	5
<input type="checkbox"/> Gradient Descent (GD):.....	5
<input type="checkbox"/> Stochastic Gradient Descent (SGD):	5
<input type="checkbox"/> Minibatch Stochastic Gradient Descent:	6
<input type="checkbox"/> RMSProp (Root Mean Square Propagation):	6
<input type="checkbox"/> Adagrad:	6
<input type="checkbox"/> Adam - Adaptive Moment Estimation:	7
c. So sánh và đánh giá các thuật toán.....	8
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.....	9
a. Tìm hiểu về Continual Learning	9
b. Tìm hiểu về Test Production	9
c. Kết hợp Continual Learning và Test Production trong việc xây dựng một mô hình học máy.....	10
TÀI LIỆU THAM KHẢO	11

NỘI DUNG BÁO CÁO

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

a. Phương pháp tối ưu (Optimizer) trong học máy là gì?

Optimizer là một thuật toán hoặc phương pháp được dùng để điều chỉnh các thuộc tính của một mạng nơ-ron hoặc bất kỳ mô hình nào khác chẳng hạn như trọng số(weights) và tốc độ học(learning rate) để giảm thiểu lỗi hoặc hàm mất mát. Mục tiêu chính là tối ưu hóa các tham số để đạt được kết quả tốt nhất.

b. Các thuật toán tối ưu trong học máy

• Gradient Descent (GD):

Gradient Descent là một trong những phương pháp tối ưu hóa phổ biến trong machine learning, được sử dụng để cập nhật các tham số của mô hình dựa trên đạo hàm của hàm mất mát.

Trong quá trình huấn luyện, Gradient Descent tính toán gradient của hàm mất mát theo các tham số và điều chỉnh các tham số đó theo một tỷ lệ học (learning rate). Thuật toán này lặp lại quá trình này cho đến khi đạt được điều kiện dừng, ví dụ như đạt đến số lượng epoch (vòng lặp) đã cho, hoặc đạt đến một giới hạn độ lỗi mong muốn.

Gradient Descent còn có các biến thể như: Stochastic Gradient Descent và Mini-batch Gradient Descent,... Mỗi hình thức này có cách tính toán gradient và cập nhật tham số khác nhau, đáp ứng cho các yêu cầu khác nhau về tốc độ hội tụ và tài nguyên tính toán.

Đối với hàm số một biến, Gradient Descent sử dụng đạo hàm của hàm số để xác định hướng và kích thước của các bước di chuyển. Quá trình này được lặp lại cho đến khi đạt được điều kiện dừng, ví dụ như đạt đủ số lần lặp hoặc đạt đến sai số mong muốn.

Đối với hàm số nhiều biến, Gradient Descent sử dụng gradient của hàm số, tức là vector các đạo hàm riêng theo từng biến, để xác định hướng và kích thước của các bước di chuyển. Tương tự như trường hợp hàm một biến, quá trình lặp lại cho đến khi đạt được điều kiện dừng.

• Stochastic Gradient Descent (SGD):

Là một biến thể của Gradient Descent trong optimizer được sử dụng trong việc tối ưu hàm mục tiêu cho các bài toán lớn hoặc có tập dữ liệu lớn.

Trong SGD, thay vì tính toán gradient trên toàn bộ tập dữ liệu, chúng ta chỉ tính toán gradient trên một mẫu ngẫu nhiên từ tập dữ liệu.

- **Minibatch Stochastic Gradient Descent:**

Là sự kết hợp giữa Batch và Stochastic Gradient Descent. Nó tính toán gradient và cập nhật tham số sau mỗi mini-batch, một phần nhỏ của tập dữ liệu. Phương pháp này kết hợp tính ổn định của Batch Gradient Descent và tốc độ hội tụ nhanh của Stochastic Gradient Descent.

- **RMSProp (Root Mean Square Propagation):**

RMSProp là một phương pháp điều chỉnh tỷ lệ học (learning rate) trong quá trình tối ưu hóa. Nó sử dụng trung bình bình phương của gradient để chuẩn hóa gradient, từ đó cân bằng kích thước bước dựa trên độ dốc của các tham số. RMSProp giúp giảm bước cho độ dốc lớn để tránh hiện tượng phát nổ độ dốc (Exploding Gradient), và tăng bước cho độ dốc nhỏ để tránh biến mất độ dốc (Vanishing Gradient). Phương pháp này tự động điều chỉnh tốc độ học tập và chọn tỷ lệ học tập phù hợp cho mỗi tham số.

$$E[g^2]_t = 0.9[g^2]_{t-1} + 0.1g_t^2$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \varepsilon}} g_t$$

- **Adagrad:**

Thực hiện giảm dần độ dốc bằng cách thay đổi tốc độ học tập.

Được cải thiện hơn bằng cách cho trọng số học tập chính xác dựa vào đầu vào trước nó để tự điều chỉnh tỉ lệ học theo hướng tối ưu nhất thay vì với một tỉ lệ học duy nhất cho tất cả các nút.

$$w_{t+1} = w_t - \frac{\eta}{\sqrt{G_t + \epsilon}} \cdot g_t$$

- **Adam - Adaptive Moment Estimation:**

Adam (Adaptive Moment Estimation) là một thuật toán tối ưu hóa cho các hàm mục tiêu ngẫu nhiên dựa trên độ dốc bậc nhất và ước tính thích ứng của các khoảng khắc bậc thấp. Đây là một phương pháp tối ưu hóa ngẫu nhiên hiệu quả chỉ yêu cầu độ dốc bậc nhất với yêu cầu bộ nhớ thấp. Phương pháp này tính toán tốc độ học thích ứng cho các tham số khác nhau. Adam sử dụng ước tính về khoảng khắc thứ nhất và thứ hai để điều chỉnh tốc độ học cho từng trọng số của mạng lưới thần kinh.

Nó kết hợp các ưu điểm của hai thuật toán tối ưu hóa AdaGrad và RMSProp để giảm dần độ dốc ngẫu nhiên. Thay vì điều chỉnh tốc độ học dựa trên mức trung bình tại thời điểm đầu tiên (giá trị trung bình) như RMSProp, Adam sử dụng mức trung bình tại thời điểm thứ hai của gradient (phương sai tập trung).

Công thức cập nhật tham số:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{v_t + \epsilon}} \cdot m_t$$

Trong đó:

θ_t là vector tham số tại thời điểm t .

η là learning rate (tốc độ học).

m_t là trung bình của gradient (theo momentum).

v_t là trung bình của bình phương gradient.

ϵ là một số nhỏ (thường được sử dụng để tránh chia cho 0).

Công thức cập nhật momentum:

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$$

Trong đó:

g_t là gradient tại thời điểm t .

β_1 là hệ số giảm dần (thường là 0.9).

Công thức cập nhật bình phương gradient:

$$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$$

Trong đó:

g_t^2 là bình phương gradient tại thời điểm t .

β_2 là hệ số giảm dần cho bình phương gradient (thường là 0.999).

Thuật toán Adam kết hợp cả momentum và bình phương gradient để điều chỉnh learning rate của mô hình dựa trên gradient của từng tham số. Việc sử dụng trung bình động lượng này giúp tăng tính ổn định và hiệu quả của quá trình tối ưu hóa, đặc biệt khi mô hình gặp phải các điểm dốc khác nhau trong không gian tham số.

c. So sánh và đánh giá các thuật toán

Các thuật toán trên đều được sử dụng để tối ưu hóa hàm mục tiêu trong quá trình huấn luyện mô hình học máy.

- Gradient Descent (GD):

Ưu điểm: Đơn giản

Nhược điểm: Tốn nhiều thời gian tính toán cho tập dữ liệu lớn..

- Mini-batch Stochastic Gradient Descent:

Ưu điểm: Hiệu suất tốt, có thể sử dụng song song trên các bản ghi.

Nhược điểm: Cần lựa chọn kích thước batch phù hợp, không phù hợp cho dữ liệu rất lớn.

- Stochastic Gradient Descent (SGD):

Ưu điểm: Đơn giản, thích hợp cho dữ liệu lớn.

Nhược điểm: Không ổn định, dễ bị mắc kẹt ở các điểm cực tiểu cục bộ.

- Adagrad

Ưu điểm: Hiệu quả với các dữ liệu thưa, tỷ lệ học tập tự điều chỉnh cho từng tham số.

Nhược điểm: Tích lũy bình phương đạo hàm có thể dẫn đến vấn đề với tỷ lệ học tập quá nhỏ.

- RMSProp

Ưu điểm: Tự điều chỉnh tốc độ học tập, phù hợp cho các bài toán với dữ liệu không đồng nhất.

Nhược điểm: Cần tinh chỉnh tham số để đạt hiệu suất tối ưu.

- Adam

Ưu điểm: Hiệu suất cao, thường hoạt động tốt trên nhiều loại vấn đề.

Nhược điểm: Cần cấu hình tham số đúng để đạt hiệu suất tối ưu.

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.

a. Tìm hiểu về Continual Learning

Học máy suốt đời (Lifelong Machine Learning: LML) hoặc Continual Learning là một quá trình học liên tục trong đó bộ học đã thực hiện một chuỗi các bài toán trước đó và sử dụng tri thức đã học để hỗ trợ giải quyết bài toán mới. Mục tiêu của LML là tối ưu hóa hiệu năng của bài toán mới và có thể áp dụng tri thức đã học để giải quyết các bài toán còn lại. Cơ sở tri thức (KB) được cập nhật với tri thức từ các bài toán trước đó. Continual Learning trong Machine Learning và AI cho phép mô hình học tập liên tục từ các luồng dữ liệu không ngừng, giúp thích ứng với sự thay đổi của dữ liệu theo thời gian.

b. Tìm hiểu về Test Production

Test Production là phương pháp đánh giá hiệu suất của mô hình trong một môi trường thay đổi. Nó đảm bảo rằng mô hình được kiểm tra và đánh giá trên dữ liệu thực tế trong quá trình triển khai. Bằng việc kiểm thử trong môi trường production, chúng ta tạo một mức độ tự tin cao hơn trong các bản release. Công ty có thể thấy cách ứng dụng phản ứng với code mới khi được đẩy vào thế giới người dùng thực. Test Production trở thành thành phần quan trọng của chiến lược chất lượng ứng dụng tương lai.

c. Kết hợp Continual Learning và Test Production trong việc xây dựng một mô hình học máy

Để áp dụng Continual Learning và Test Production vào một bài toán cụ thể, ta có thể xem xét ví dụ về hệ thống gợi ý phim. Trong bài toán này, hệ thống được xây dựng để đề xuất các bộ phim dựa trên sở thích của người dùng.

Với Continual Learning, hệ thống có thể tiếp tục học tập và cập nhật kiến thức về sở thích của người dùng khi họ tương tác với hệ thống. Điều này đảm bảo rằng hệ thống luôn cung cấp các đề xuất phim mới nhất và phù hợp với người dùng.

Đối với Test Production, chúng ta có thể tạo ra các bộ kiểm tra để đánh giá hiệu suất của hệ thống gợi ý phim. Các bộ kiểm tra này có thể được thiết kế để đo lường độ chính xác của hệ thống trong việc đề xuất các bộ phim phù hợp với sở thích của người dùng. Đồng thời, chúng ta cũng cần đảm bảo rằng các bộ kiểm tra này phản ánh đúng phân phối của dữ liệu thực tế mà người dùng có thể gặp phải.

TÀI LIỆU THAM KHẢO

1. <https://www.phamduytung.com/blog/2021-01-15---adabelief-optimizer/>
2. <https://viblo.asia/p/optimizer-hieu-sau-ve-cac-thuat-toan-toi-uu-gdsgdadam-Qbq5QQ9E5D8>
3. http://www.uet.vnu.edu.vn/~thuyhq/Student_Thesis/K59_Nguyen-Thi-Lan_Thesis.pdf

<https://viblo.asia/p/tai-sao-ban-nen-kiem-thu-tren-moi-truong-production-WAyK8oOm5xX>