

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



TRƯỜNG HUỖNH ĐẶNG KHOA- 521H0503

OPTIMIZATION METHODS IN MACHINE LEARNING

FINAL EXAM REPORT

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**



TRƯỜNG HUỲNH ĐĂNG KHOA - 521H0503

OPTIMIZATION METHODS IN MACHINE LEARNING

FINAL EXAM REPORT

1.1 Tìm hiểu, so sánh các phương pháp Optimizer trong huấn luyện mô hình học máy	4
1. Gradient Descent	4
2. Gradient Descent with Momentum (GDM)	5
3. Adagrad	5
4. RMSProp (Root Mean Squared Propagation)	6
5. Adam	6
1.2 Tìm hiểu về Continual Learning và Test Production khi xây dựng một giải pháp học máy để giải quyết một bài toán nào đó.	7
1. Học liên tục (Continual Learning):	7
2. Test Production	8

Bài 1 (3 điểm):

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.

Trả lời:

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

Trong huấn luyện học máy, tối ưu hóa (optimizing) là một phần quan trọng để điều chỉnh các tham số của mô hình sao cho mô hình đạt được hiệu suất tốt nhất trên dữ liệu huấn luyện. Dưới đây là một số phương pháp Optimizer trong huấn luyện mô hình học máy.

1. Gradient Descent

Gradient Descent là một phương pháp tối ưu được sử dụng để điều chỉnh tham số của một mô hình học máy để giảm thiểu một hàm mất mát (loss function). Gradient Descent có nghĩa là giảm dần độ dốc, có nghĩa là theo phương pháp này, ta sẽ chọn 1 nghiệm ngẫu nhiên và cho vào vòng lặp (epoch). Sau mỗi vòng lặp, tham số sẽ được cập nhật cho đến khi đạt tới điều kiện dừng (ví dụ điều kiện dừng là độ chênh lệch của loss function nhỏ nhất).

Ưu điểm:

- Thuật toán Gradient Descent là một thuật toán dễ học, dễ hiểu và dễ áp dụng.
- Đây là một thuật toán phổ biến và được sử dụng nhiều để giải quyết các vấn đề học máy và học sâu.

Nhược điểm:

- Phụ thuộc vào tỷ lệ học (learning rate). Tỷ lệ học lớn thì có thể dẫn đến sự không hội tụ, tỉ lệ học nhỏ thì có thể làm chậm quá trình học.
- Nếu hàm mất mát có nhiều điểm cực tiểu cục bộ, thuật toán Gradient Descent có thể cho ra kết quả rơi vào những điểm đó thay vì tìm ra cực tiểu toàn cục.

2. Gradient Descent with Momentum (GDM)

Đây là một biến thể của Gradient Descent được thiết kế để giúp tăng tốc quá trình hội tụ và giảm tốc độ dao động của thuật toán khi tiến gần đến điểm cực tiểu. Giả sử, khi tới được điểm cực tiểu cục bộ, thay vì dừng lại, thì nó sẽ được đẩy đi tiếp cho đến khi đến được điểm cực tiểu toàn cục.

Ưu điểm:

- Giảm giao động của quá trình tối ưu hóa, giúp các thuật toán hội tụ nhanh và ổn định hơn.
- Giải quyết được vấn đề của Gradient Descent là không thể tới được điểm cực tiểu toàn cục mà dừng lại ở điểm cực tiểu cục bộ

Nhược điểm:

- Tuy có thể giúp thuật toán đi tới điểm cực tiểu toàn cục, tuy nhiên vẫn có khả năng bị Overshooting khi đà quá cao, dẫn tới việc dao động qua lại ở điểm cực tiểu trước khi dừng lại hẳn.
- Khả năng dừng ở những điểm có đạo hàm bằng 0 nhưng không phải điểm cực tiểu.

3. Adagrad

Là một thuật toán tối ưu trong học máy, nhưng khác với các thuật toán trước, Adagrad có thể coi learning rate như một tham số và điều chỉnh cho mỗi tham số dựa trên mức độ quan trọng của tham số đó đối với mô hình.

Ưu điểm:

- Adagrad có thể tự điều chỉnh learning rate cho từng tham số, chỉ cần để default là 0.01. Điều này hóa tỷ lệ học cho từng tham số một cách hiệu quả.

Nhược điểm:

- Tăng độ phức tạp của tính toán, làm chậm quá trình học.

4. RMSProp (Root Mean Squared Propagation)

Sử dụng trung bình bình phương của gradient để chuẩn hóa gradient. Có tác dụng cân bằng kích thước bước - giảm tốc độ của gradient lớn để tránh hiện tượng phát nổ độ dốc (Exploding Gradient), và giúp tăng tốc quá trình hội tụ của thuật toán. Nó bao gồm một số thuộc tính của Adagrad, đồng thời giải quyết một số vấn đề mà Adagrad có thể gặp trong lúc huấn luyện

Ưu điểm:

- Tự động điều chỉnh learning rate cho từng tham số, điều này giúp nó hoạt động hiệu quả với các bài toán có gradient thay đổi đáng kể
- Khắc phục được vấn đề học chậm dần của Adagrad

Nhược điểm:

- Có khả năng dừng sớm
- Thuật toán có thể cho ra nghiệm chỉ là local minimum chứ không được global minimum như GDM.

5. Adam

Là thuật toán kết hợp ý tưởng giữa GDM và RMSProp, để tận dụng ưu điểm của cả hai. Adam được thiết kế để cung cấp một phương pháp tối ưu hóa hiệu quả, tự điều chỉnh learning rate và giảm giao động của quá trình tối ưu hóa

Ưu điểm:

1.2 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. Học liên tục (Continual Learning):

- Học liên tục (Continual Learning) là một phương pháp học mà mô hình học máy học được cập nhật liên tục dữ liệu mới mà không cần huấn luyện lại.
- Mục tiêu là duy trì học và mở rộng kiến thức của mô hình với dữ liệu mới mà không quên hay ghi nhớ quá mức về dữ liệu cũ

Các phương pháp học liên tục:

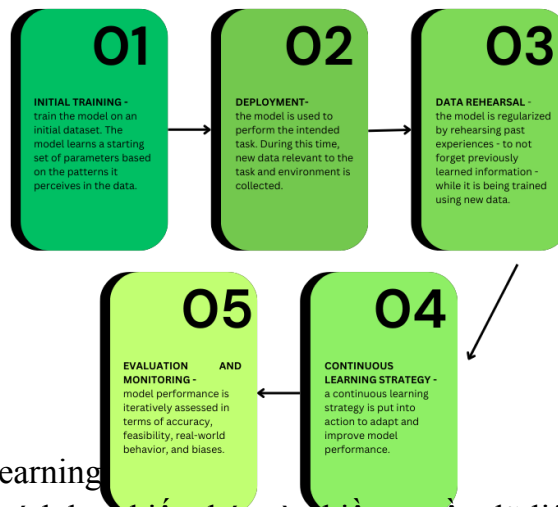
- Elastic Weight Consolidation (EWC):
 - Đây là một kỹ thuật giúp giữ trọng số của mô hình ổn định khi đang học thêm dữ liệu mới. Nó giúp tránh quên các kiến thức đã học trước đó
- Replay Buffer:
 - Lưu trữ một lượng lớn các ví dụ từ quá khứ để mô hình có thể đào tạo lại trên dữ liệu cũ khi cần
- Progressive Neural Networks (PNN):
 - Là một kiến trúc mạng nơ-rôn được thiết kế để mở rộng kiến thức mà không làm ảnh hưởng quá mức tới kiến thức đã học.

- Mỗi lớp mới được thêm vào để xử lý dữ liệu mới mà không làm thay đổi trọng số của các lớp cũ
- Meta-Learning:
 - Mô hình được đào tạo để nhanh chóng học được dữ liệu mới bằng cách sử dụng kinh nghiệm từ việc học các dữ liệu trước đó.

Quá trình học liên tục:

Học liên tục gồm 5 bước:

The Continuous Learning Process



Ưu điểm của continual learning

- Học liên tục giúp tích hợp kiến thức từ nhiều nguồn dữ liệu và nhiệm vụ khác nhau vào một mô hình duy nhất
- Giúp giảm thiểu Catastrophic Forgetting, giảm nguy cơ quên kiến thức cũ khi nạp thêm kiến thức mới
- Mô hình có khả năng thích ứng nhanh chóng với sự thay đổi của môi trường và dữ liệu
- Sử dụng Replay Buffer giúp duy trì sự đa dạng của dữ liệu đã được nạp vào, giúp mô hình có thể học nhiều từ các tình huống khác nhau

Nhược điểm:

- Có thể gặp tình trạng quên đột ngột (Catastrophic Forgetting) khi trọng số của mô hình được điều chỉnh để phù hợp với dữ liệu mới và làm mất đi kiến thức đã học của dữ liệu cũ.
- Mô hình có thể trở nên phức tạp do việc tích hợp kiến thức từ nhiều nguồn và nhiều nhiệm vụ khác nhau, làm tăng độ phức tạp của mô hình và đòi hỏi nhiều nguồn tài nguyên.

2. Test Production

Test production là quá trình kiểm thử mô hình học máy sau khi nó đã được huấn luyện. Việc này bao gồm đánh giá hiệu suất của mô hình trên dữ liệu mới mà nó chưa thấy trong quá trình huấn luyện. Quá trình này quan trọng để đảm bảo rằng mô hình có khả năng khái quát hóa tốt mà không chỉ đơn thuần là nhớ dữ liệu huấn luyện.

Các bước thực hiện kiểm thử:

1. Chuẩn bị dữ liệu kiểm thử: Chuẩn bị dữ liệu mà mô hình chưa bao giờ thấy. Dữ liệu này nên phản ánh thực tế mà mô hình có thể gặp khi triển khai.
2. Tiền xử lý dữ liệu: Áp dụng các bước tiền xử lý dữ liệu giống trong quá trình huấn luyện để đảm bảo tính nhất quán và chất lượng của dữ liệu dùng cho quá trình test.
3. Chạy mô hình trên dữ liệu kiểm thử
4. Đánh giá hiệu suất mô hình, phân tích kết quả và triển khai vào môi trường thực tế.