TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**HỒ MINH CHÍ TÂN – 519H0044**

**BÁO CÁO CUỐI KÌ**

**NHẬP MÔN HỌC MÁY**

**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**



**HỒ MINH CHÍ TÂN – 519H0044**

**BÁO CÁO CUỐI KỲ**

**NHẬP MÔN HỌC MÁY**

Người hướng dẫn

**TS. Lê Anh Cường**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

**LỜI CẢM ƠN**

Chúng em xin chân thành cảm ơn …………………………………… …………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

*TP. Hồ Chí Minh, ngày ... tháng … năm 20..*

*Tác giả*

*(Ký tên và ghi rõ họ tên)*

**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 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 Dự 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 Dự 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 20..*

*Tác giả*

*(Ký tên và ghi rõ họ tên)*

**MỤC LỤC**

[CHƯƠNG 1. PHƯƠNG PHÁP OPTIMIZER 2](#_Toc154125500)

[1.1 Khái niệm Optimizer 2](#_Toc154125501)

[1.2 Ví dụ sử dụng thuật toán tối ưu trong Machine Learning 2](#_Toc154125502)

[1.3 Gradient Descent 3](#_Toc154125503)

[1.4 Stochastic Gradient Descent 3](#_Toc154125504)

[1.5 Mini-batch Gradient Descent 4](#_Toc154125505)

[1.6 Adam (Adaptive Moment Estimation) 5](#_Toc154125506)

[1.7 Adagrad (Adaptive Gradient Algorithm) 6](#_Toc154125507)

[1.8 So sánh các phương pháp Optimizer 6](#_Toc154125508)

[CHƯƠNG 2. CONTINUAL LEARNING VÀ TEST PRODUCTION 9](#_Toc154125509)

[2.1 Continual Learning 9](#_Toc154125510)

[2.1.1 Các phương pháp của Continual Learning 9](#_Toc154125511)

[2.2 Test Production 10](#_Toc154125512)

[2.2.1 Các kỹ thuật và chiến lược phổ biến 10](#_Toc154125513)

[2.3 Áp dụng Continual Learning và Test Production 12](#_Toc154125514)

# PHƯƠNG PHÁP OPTIMIZER

## Khái niệm Optimizer

Thuật toán tối ưu là cơ sở để xây dựng mô hình neural network với mục đích "học " được các features ( hay pattern) của dữ liệu đầu vào, từ đó có thể tìm 1 cặp weights và bias phù hợp để tối ưu hóa model

## Ví dụ sử dụng thuật toán tối ưu trong Machine Learning

Phân loại hình ảnh:

* Tập dữ liệu gồm các hình ảnh của các loại hoa khác nhau.
* Mục tiêu: Xây dựng mô hình phân loại có thể phân loại chính xác các loại hoa trong tập dữ liệu này.
* Tham số: Các trọng số của mô hình.
* Mục đích của thuật toán tối ưu: Tìm ra các trọng số này sao cho xác suất dự đoán của mô hình là cao nhất.

Hồi quy

* Tập dữ liệu gồm các điểm dữ liệu (x, y).
* Mục tiêu: Xây dựng mô hình hồi quy có thể dự đoán giá trị y dựa trên giá trị x.
* Tham số: Các hệ số của mô hình.
* Mục đích của thuật toán tối ưu: Tìm ra các hệ số này sao cho sai số dự đoán của mô hình là nhỏ nhất.

Học sâu (Deep Learning)

* Tập dữ liệu gồm các điểm dữ liệu.
* Mục tiêu: Xây dựng mô hình học máy sâu có thể học được các đặc trưng từ dữ liệu và đưa ra các dự đoán chính xác.
* Tham số: Các trọng số và bias của mô hình.
* Mục đích của thuật toán tối ưu: Tìm ra các trọng số và bias này sao cho mô hình có thể học được các đặc trưng từ dữ liệu và đưa ra các dự đoán chính xác.

## Gradient Descent

Gradient Descent (GD) là một thuật toán tối ưu hóa được sử dụng để cập nhật trọng số của mô hình máy học, nhằm giảm thiểu hàm mất mát (loss function). Mục tiêu của Gradient Descent là tìm ra bộ trọng số mà hàm mất mát đạt giá trị nhỏ nhất

Dưới đây là cách thuật toán Gradient Descent hoạt động:

1. Khởi tạo trọng số: Bắt đầu với một bộ trọng số ngẫu nhiên hoặc được chọn trước.
2. Dự đoán: Sử dụng bộ trọng số hiện tại để thực hiện dự đoán trên tập dữ liệu.
3. Tính toán độ lệch (gradient): Tính toán đạo hàm của hàm mất mát đối với từng trọng số. Đạo hàm này chỉ ra hướng và mức độ cần điều chỉnh trọng số để giảm mất mát.
4. Cập nhật trọng số: Di chuyển theo đối phương của đạo hàm với một bước nhỏ, được xác định bởi một tham số gọi là learning rate. Công thức cập nhật trọng số có thể được mô tả như sau:

new weight = old weight − learning rate × gradient

1. Lặp lại: Lặp lại quá trình từ bước 2 đến bước 4 cho đến khi một điều kiện dừng được đáp ứng

## Stochastic Gradient Descent

Stochastic Gradient Descent (SGD) là một biến thể của thuật toán Gradient Descent (GD) được sử dụng để tối ưu hóa mô hình máy học. Trong SGD, không giống như GD, việc cập nhật trọng số không được thực hiện sau mỗi epoch (quá trình đi qua toàn bộ tập dữ liệu huấn luyện), mà thay vào đó được thực hiện sau mỗi mẫu dữ liệu.

Dưới đây là cách SGD hoạt động:

1. Khởi tạo trọng số: Bắt đầu với một bộ trọng số ngẫu nhiên hoặc được chọn trước.
2. Lặp lại qua dữ liệu: Đối với mỗi mẫu dữ liệu trong tập huấn luyện:
3. Dự đoán: Sử dụng bộ trọng số hiện tại để thực hiện dự đoán cho mẫu dữ liệu đó.
4. Tính toán độ lệch (gradient): Tính toán đạo hàm của hàm mất mát đối với từng trọng số, dựa trên mẫu dữ liệu hiện tại.
5. Cập nhật trọng số: Di chuyển theo đối phương của đạo hàm với một bước nhỏ, được xác định bởi một tham số learning rate.
6. Lặp lại quá trình: Lặp lại quá trình trên cho đến khi một điều kiện dừng được đáp ứng

## Mini-batch Gradient Descent

Mini-batch Gradient Descent là một biến thể của thuật toán Gradient Descent (GD) và là sự kết hợp giữa GD và Stochastic Gradient Descent (SGD). Thay vì cập nhật trọng số sau mỗi epoch như GD hoặc sau mỗi mẫu dữ liệu như SGD, Mini-batch Gradient Descent cập nhật trọng số sau mỗi mini-batch, là một nhóm nhỏ các mẫu dữ liệu được lựa chọn từ toàn bộ tập dữ liệu.

Dưới đây là cách Mini-batch Gradient Descent hoạt động:

1. Chia tập dữ liệu thành mini-batches: Tập dữ liệu huấn luyện được chia thành các mini-batches có kích thước nhỏ. Các mini-batches này là một phần của tập dữ liệu lớn.
2. Khởi tạo trọng số: Bắt đầu với một bộ trọng số ngẫu nhiên hoặc được chọn trước.
3. Lặp lại qua các mini-batches: Đối với mỗi mini-batch:

* Dự đoán: Sử dụng bộ trọng số hiện tại để thực hiện dự đoán cho mini-batch đó.
* Tính toán độ lệch (gradient): Tính toán đạo hàm của hàm mất mát đối với từng trọng số, dựa trên mini-batch hiện tại.
* Cập nhật trọng số: Di chuyển theo đối phương của đạo hàm với một bước nhỏ, được xác định bởi một tham số learning rate.

1. Lặp lại quá trình: Lặp lại quá trình trên cho đến khi một điều kiện dừng được đáp ứng.

## Adam (Adaptive Moment Estimation)

Adam (Adaptive Moment Estimation) là một thuật toán tối ưu hóa được sử dụng để cập nhật trọng số của mô hình máy học trong quá trình huấn luyện. Thuật toán này kết hợp thông tin từ cả mean (trung bình) và variance (phương sai) của gradient để điều chỉnh learning rate cho từng tham số riêng lẻ.

Dưới đây là cách thuật toán Adam hoạt động:

1. Khởi tạo trọng số và các giá trị ban đầu:

* Bắt đầu với bộ trọng số ngẫu nhiên hoặc được chọn trước.
* Khởi tạo hai bộ nhớ chứa mean và variance của gradient, ban đầu đặt giá trị của chúng bằng 0.

1. Lặp lại qua dữ liệu: Đối với mỗi mini-batch:

* Tính toán gradient: Tính toán đạo hàm của hàm mất mát đối với từng trọng số, dựa trên mini-batch hiện tại.
* Cập nhật mean và variance: Tính toán mean và variance của gradient sử dụng exponential moving averages.
* Điều chỉnh mean và variance đạo hàm: Bình phương mỗi thành phần của gradient và điều chỉnh chúng bằng mean và variance tính được ở bước trước.
* Cập nhật trọng số: Di chuyển theo đối phương của đạo hàm đã được điều chỉnh, với một bước nhỏ được xác định bởi learning rate.

1. Lặp lại quá trình: Lặp lại quá trình trên cho đến khi một điều kiện dừng được đáp ứng.

## Adagrad (Adaptive Gradient Algorithm)

Adagrad (Adaptive Gradient Algorithm) là một thuật toán tối ưu hóa được sử dụng để cập nhật trọng số của mô hình máy học trong quá trình huấn luyện. Thuật toán này có khả năng tự điều chỉnh learning rate của từng tham số dựa trên lịch sử của các gradient đã xuất hiện.

Dưới đây là cách Adagrad hoạt động:

1. Khởi tạo trọng số và các giá trị ban đầu:

* Bắt đầu với bộ trọng số ngẫu nhiên hoặc được chọn trước.
* Khởi tạo một bộ nhớ chứa tổng bình phương của các gradient cho mỗi tham số, ban đầu đặt giá trị của chúng bằng 0.

1. Lặp lại qua dữ liệu:

* Tính toán gradient: Tính toán đạo hàm của hàm mất mát đối với từng trọng số, dựa trên mini-batch hiện tại.
* Cập nhật tổng bình phương của gradient: Thêm bình phương của gradient (element-wise) vào tổng bình phương đã tính trước đó.
* Cập nhật trọng số: Di chuyển theo đối phương của đạo hàm, chia cho căn bậc hai của tổng bình phương của gradient, nhân với một learning rate.

1. Lặp lại quá trình: Lặp lại quá trình trên cho đến khi một điều kiện dừng được đáp ứng.

## So sánh các phương pháp Optimizer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Thuật toán** | **Mô tả** | **Ưu điểm** | **Nhược điểm** | **Phù hợp với** |
| Gradient Descent (GD) | Cập nhật tham số theo hướng giảm dần của độ dốc hàm mất mát toàn bộ tập dữ liệu. | Đơn giản, dễ hiểu, hội tụ tốt cho hàm phức lồi. | Chậm hội tụ cho tập dữ liệu lớn, nhạy cảm với kích thước học tập (learning rate). | Bài toán nhỏ, hàm phức lồi |
| Stochastic Gradient Descent (SGD) | Cập nhật tham số theo hướng giảm dần của độ dốc hàm mất mát tại một mẫu dữ liệu ngẫu nhiên | Nhanh hơn GD cho tập dữ liệu lớn, thoát khỏi điểm cực tiểu cục bộ dễ hơn. | Biến động lớn, có thể dao động quanh cực tiểu, hội tụ chậm hơn GD cho hàm phức lồi. | Tập dữ liệu lớn, thoát khỏi điểm cực tiểu cục bộ, chấp nhận độ nhiễu cao. |
| Mini-batch Gradient Descent (Minibatch GD) | Cập nhật tham số theo hướng giảm dần của độ dốc hàm mất mát trên một nhóm nhỏ mẫu dữ liệu ngẫu nhiên | Cân bằng giữa GD và SGD, nhanh hơn GD, ổn định hơn SGD, ít nhiễu hơn SGD. | Tốn thêm bộ nhớ để lưu trữ mini-batch. | Tập dữ liệu lớn, cân bằng giữa tốc độ và ổn định. |
| Adam (Adaptive Moment Estimation) | Kết hợp SGD với momentum và learning rate. | Nhanh hội tụ, ổn định, tự điều chỉnh learning rate, ít nhạy cảm với kích thước học tập. | Tốn thêm bộ nhớ lưu trữ các momen, phức tạp hơn SGD. | Bài toán phức tạp, nhiều tham số, cần tốc độ hội tụ nhanh và ổn định. |
| Adagrad (Adaptive Gradient Algorithm) | Điều chỉnh learning rate riêng cho từng tham số dựa trên lịch sử cập nhật. | Hiệu quả tốt cho các tham số có gradient thay đổi theo thời gian, giảm ảnh hưởng của gradient lớn nhiễu. | Có thể tốn kém về tính toán cho tập dữ liệu lớn, hội tụ chậm hơn Adam. | Bài toán có các tham số với gradient thay đổi theo thời gian, nhạy cảm với gradient nhiễu lớn. |

# CONTINUAL LEARNING VÀ TEST PRODUCTION

## Continual Learning

Continual Learning là một chiến lược học máy cho phép mô hình không ngừng học và cải thiện hiệu suất của nó theo thời gian, ngay cả khi môi trường hoặc dữ liệu đầu vào thay đổi. Điều này thường được thực hiện thông qua sử dụng các thuật toán học máy chuyên dụng như incremental learning, online learning, hoặc reinforcement learning.

### Các phương pháp của Continual Learning

Elastic Weight Consolidation (EWC):

* Mô tả: EWC giữ lại các trọng số quan trọng cho các nhiệm vụ trước đó bằng cách thêm một thành phần "kiểm soát" vào hàm mất mát.
* Ứng dụng: Phù hợp khi cần giữ lại thông tin quan trọng từ nhiệm vụ cũ khi học từ nhiệm vụ mới.

Online Learning và Stochastic Gradient Descent (SGD):

* Mô tả: Online Learning liên quan đến việc mô hình liên tục cập nhật trọng số khi dữ liệu mới xuất hiện, thường sử dụng các biến thể của SGD.
* Ứng dụng: Phù hợp trong các tình huống yêu cầu sự động đà và nhanh chóng trong việc học từ dữ liệu mới.

Meta-learning:

* Mô tả: Mô hình được đào tạo để học cách nhanh chóng học từ dữ liệu mới, thường thông qua việc mô phỏng nhiều tình huống học khác nhau.
* Ứng dụng: Phù hợp trong các tình huống yêu cầu mô hình học nhanh và hiệu quả từ dữ liệu mới.

Dynamic Architectures và Expandable Networks:

* Mô tả: Mô hình có thể tự điều chỉnh kích thước hoặc cấu trúc của mình khi gặp dữ liệu mới.
* Ứng dụng: Phù hợp khi cần mô hình có khả năng mở rộng và thích ứng với sự biến đổi của dữ liệu.

Memory Augmented Networks:

* Mô tả: Sử dụng bộ nhớ ngoại vi để lưu giữ thông tin quan trọng từ nhiệm vụ cũ và sử dụng lại nó khi học từ dữ liệu mới.
* Ứng dụng: Hiệu quả trong việc giữ lại thông tin quan trọng qua thời gian.

## Test Production

Test Production là một phần quan trọng trong quá trình xây dựng giải pháp học máy để đảm bảo rằng mô hình không chỉ hiệu quả trên tập dữ liệu kiểm thử mà bạn đã sử dụng trong quá trình đào tạo, mà còn là hiệu quả và đáng tin cậy trên dữ liệu thực tế và trong môi trường triển khai.

### Các kỹ thuật và chiến lược phổ biến

**Kiểm Thử Tự Động:**

* Mô tả: Tự động hóa quy trình kiểm thử để nhanh chóng và đảm bảo tính toàn vẹn của các bài kiểm tra.
* Ưu điểm: Tiết kiệm thời gian, giảm sai sót do con người, có thể thực hiện kiểm thử liên tục.
* Hành động: Sử dụng các framework kiểm thử tự động như Selenium cho ứng dụng web, PyTest hoặc unittest cho mã nguồn Python.

**Kiểm Thử Đa Dạng Dữ Liệu:**

* Mô tả: Sử dụng tập dữ liệu kiểm thử đa dạng, đại diện cho nhiều trường hợp sử dụng và điều kiện khác nhau.
* Ưu điểm: Đảm bảo tính toàn vẹn của mô hình trên nhiều trường hợp sử dụng và giả mạo điều kiện thực tế.
* Hành động: Xác định các tình huống biên, tình huống cực đại, và sử dụng dữ liệu kiểm thử đa dạng.

**Kiểm Thử Độ Bền (Endurance Testing):**

* Mô tả: Kiểm tra khả năng của mô hình hoạt động hiệu quả trong thời gian dài và trong điều kiện áp lực.
* Ưu điểm: Đảm bảo rằng mô hình có thể duy trì hiệu suất và tính ổn định dưới tải công việc liên tục.
* Hành động: Thực hiện kiểm thử liên tục trong khoảng thời gian dài và đo lường hiệu suất.

**Kiểm Thử An Toàn (Security Testing):**

* Mô tả: Kiểm tra mô hình để đảm bảo rằng nó không bị mối đe dọa về an ninh và bảo mật.
* Ưu điểm: Đảm bảo rằng mô hình không bị tổn thương và duy trì sự bảo mật của dữ liệu.
* Hành động: Sử dụng các công cụ kiểm thử an ninh, kiểm tra tình trạng bảo mật của mô hình.

**Kiểm Thử Độ Tin Cậy (Reliability Testing):**

* Mô tả: Đảm bảo rằng mô hình hoạt động đúng đắn và đáng tin cậy trong mọi điều kiện.
* Ưu điểm: Xác định và giải quyết lỗi hoặc vấn đề liên quan đến sự đáng tin cậy của mô hình.
* Hành động: Thực hiện kiểm thử độ tin cậy với nhiều điều kiện khác nhau.

**Kiểm Thử Tương Thích (Compatibility Testing):**

* Mô tả: Kiểm tra khả năng tương thích của mô hình trên nhiều môi trường và thiết bị khác nhau.
* Ưu điểm: Đảm bảo rằng mô hình hoạt động đúng trên nhiều nền tảng và thiết bị.
* Hành động: Thực hiện kiểm thử trên nhiều trình duyệt, hệ điều hành và thiết bị.

**Kiểm Thử Tương Tác Hệ Thống (System Integration Testing):**

* Mô tả: Đảm bảo rằng mô hình tương tác đúng đắn với các thành phần khác trong hệ thống tổng thể.
* Ưu điểm: Đảm bảo rằng mô hình hoạt động chính xác và tương thích với các thành phần khác.
* Hành động: Thực hiện kiểm thử tích hợp với các thành phần hệ thống liên quan

## Áp dụng Continual Learning và Test Production

**Áp dụng Continual Learning:**

Bước 1: Đào tạo Ban Đầu

* Mục tiêu: Huấn luyện mô hình để nhận diện người và vật thể từ dữ liệu camera.
* Hành động: Sử dụng một tập dữ liệu lớn chứa hình ảnh của người và vật thể để đào tạo mô hình.

Bước 2: Triển Khai Mô Hình Ban Đầu

* Mục tiêu: Sử dụng mô hình để nhận diện và giám sát an ninh.
* Hành động: Triển khai mô hình để theo dõi và phát hiện người và vật thể trong môi trường thực tế.

Bước 3: Continual Learning cho dữ liệu mới

* Mục tiêu: Học từ dữ liệu mới để cải thiện khả năng nhận diện.
* Hành động: Khi hệ thống tiếp tục hoạt động, liên tục thu thập dữ liệu mới từ camera và sử dụng chúng để huấn luyện mô hình tiếp theo một cách liên tục.

Bước 4: Bảo toàn Kiến Thức Cũ

* Mục tiêu: Đảm bảo rằng mô hình không quên kiến thức cũ khi học từ dữ liệu mới.
* Hành động: Sử dụng các phương pháp như Elastic Weight Consolidation (EWC) để giữ lại trọng số quan trọng từ mô hình ban đầu khi huấn luyện mô hình mới.

**Áp dụng Test Production:**

Bước 1: Kiểm Thử Hiệu Suất Ban Đầu

* Mục tiêu: Đánh giá hiệu suất của mô hình trên dữ liệu kiểm thử đa dạng.
* Hành động: Sử dụng tập dữ liệu kiểm thử đa dạng chứa các tình huống khác nhau như ánh sáng yếu, góc chụp khác nhau, và điều kiện thời tiết đổi lạnh ấm.

Bước 2: Kiểm Thử An Toàn và Độ Tin Cậy

* Mục tiêu: Đảm bảo rằng mô hình không bị tổn thương và hoạt động đúng trong điều kiện an toàn.
* Hành động: Kiểm thử mô hình trên tình huống như mô phỏng vật cản, ánh sáng mạnh đột ngột, hoặc nhiễu.

Bước 3: Kiểm Thử Tương Thích và Độ Bền

* Mục tiêu: Đảm bảo mô hình tương thích và duy trì hiệu suất trên nhiều thiết bị và trong thời gian dài.
* Hành động: Thực hiện kiểm thử trên các loại camera khác nhau và đảm bảo rằng mô hình duy trì hiệu suất sau một thời gian dài.

Bước 4: Kiểm Thử Tương Tác Hệ Thống và Độ Tin Cậy

* Mục tiêu: Đảm bảo rằng mô hình tương tác đúng với hệ thống tổng thể và đáp ứng đúng với thông điệp và các yêu cầu của hệ thống.
* Hành động: Thực hiện kiểm thử tích hợp với hệ thống giám sát an ninh tổng thể và đảm bảo sự tương thích.

TÀI LIỆU THAM KHẢO