TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CUỐI KÌ**

**Machine Learning**

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

*Người thực hiện*: **HUỲNH TRẦN MINH TIẾN - 520H0583**

Lớp **: 20H050202**

Khoá  **: 24**

**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**

**Continual Learning và Test Production trong học máy**

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

Người thực hiện: **HUỲNH TRẦN MINH TIẾN**

Lớp **: 20H050202**

Khoá  **: 24**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

LỜI CẢM ƠN

Em muốn bày tỏ lòng biết ơn chân thành của mình đối với thầy Lê Anh Cường vì đã cung cấp cho em điều kiện thực hiện và hoàn thành bài báo cáo giữa kỳ này một cách công bằng, minh bạch và tốt nhất.

Rất mong các thầy cô bỏ qua những sai sót có thể xảy ra trong quá trình thực hiện và hoàn thành báo cáo này. Đồng thời, em rất mong các GV bộ môn sẽ chấm công tâm và công bằng để bài báo cáo đạt điểm tốt nhất. Điều này là do cả trình độ lý luận và kinh nghiệm làm báo cáo hạn chế.

Tôi muốn bày tỏ lòng biết ơn của mình một lần nữa đến thầy

*TP. Hồ Chí Minh, ngày 23 tháng 12 năm 2023*

*Tác giả*

*Huỳnh Trần Minh Tiến*

**ĐỒ ÁN ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là sản phẩm đồ án của riêng tôi và được sự hướng dẫn của PGS.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 đồ 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 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 23 tháng 12 năm 2023*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Huỳnh Trần Minh Tiến*

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

[LỜI CẢM ƠN 1](#_Toc20341)

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN 3](#_Toc21256)

[MỤC LỤC 4](#_Toc32412)

[DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ 5](#_Toc20392)

[CHƯƠNG 1 – MỞ ĐẦU 6](#_Toc15649)

[1.1 Optimizer 6](#_Toc32526)

[1.1.1 Gradient Descent (GD) 6](#_Toc21935)

[1.1.2 Stochastic Gradient Descent (SGD) 11](#_Toc2628)

[1.1.3 Momentum 13](#_Toc23051)

[1.1.4 Adagrad 15](#_Toc7510)

[1.1.5 RMSprop 17](#_Toc2561)

[1.1.6 Adam 19](#_Toc29960)

[1.1.7 Tổng kết 21](#_Toc25291)

[CHƯƠNG 2 – Continual Learning và Test Production khi xây dựng giải pháp học máy 22](#_Toc1833)

[2.1 Continual Learning 22](#_Toc8846)

[2.2 Test Production 24](#_Toc656)

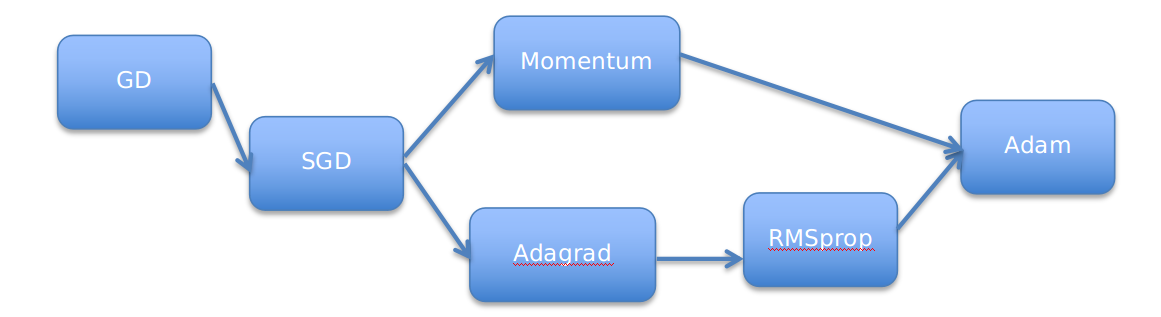
CHƯƠNG 1 – MỞ ĐẦU

* 1. Optimizer

Optimizer (tối ưu) trong quá trình huấn luyện mô hình là các thuật toán hay phương thức được sử dụng để giảm thiểu mức độ lỗi trong hàm mất mát (Loss Function), giúp mô hình cho ra được hiệu quả tốt nhất trong việc đưa ra dự đoán hoặc phân loại dữ liệu. Các thuật toán tối ưu được xem là cơ sở để xây dựng mô hình mạng Neural Network - NN (mô hình mạng nơ-ron nhân tạo mô phỏng con người) phụ thuộc vào tham số của mô hình cụ thể là Weight - ω(Trọng số) và Bias -Ь(Nguỡng) để có thể điều chỉnh trọng số và tốc độ học sau mỗi lần huấn luyện để tìm ra tập (ω,Ь) tối ưu hoặc thỏa đáng.

Một số thuật toán Optimizer thông dụng:

* Gradient Descent (GD)
* Stochastic Gradient Descent (SGD)
* Momentum
* Adagrad
* RMSprop
* Adam



1.1 Optimizer Algorithms 1

* + 1. Gradient Descent (GD)

Với ý tưởng tìm nghiệm mà điểm đó gần nhất với cực tiểu của đạo hàm số thay vì tìm giá trị nhỏ nhất của hàm số khi đạo hàm vì trong một số trường hợp điều đó là bất khả thi. Gradient Descent là thuật toán tối ưu vòng lặp (iterative optimization) thường được sử dụng cho tối ưu các bài toán lồi - Convex Optimization bằng cách tìm tập các biến nội tại (internal parameters).

Một số khái niệm trong thuật toán:

1. Gradient: tỷ lệ nghiêng/độ dốc (rate of inclination or declination of a slope), là đạo hàm của số tương ứng với mỗi biến, thuộc tính (feature/pattern) của dữ liệu đầu vào. Trong hàm số đơn biến, sử dụng khái niệm Derivative thay vì Gradient.
2. Descent: giảm dần

Linear Regression sẽ có công thức tổng quát như sau:

y = f(x) = + x

Thuật toán Gradient Descent thực thi theo trình tự:

1. Khởi tạo biến nội tại (ngẫu nhiên)
2. Đánh giá model dựa vào biến nội tại và hàm mất mát (Loss Function), với Loss sẽ được tính toán theo công thức:

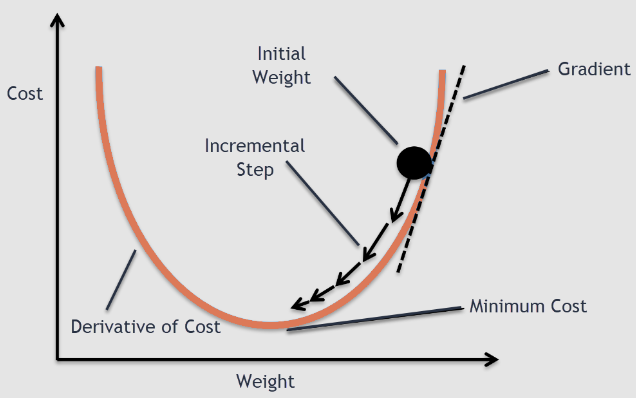
Loss = L = =

= 0, = 0

1. Cập nhật biến nội tại theo hướng tối ưu hàm mất mát (finding optimal points)
2. Lặp lại bước 2, 3 cho đến khi thỏa mãn điều kiện (vòng lặp, tỷ lệ học)

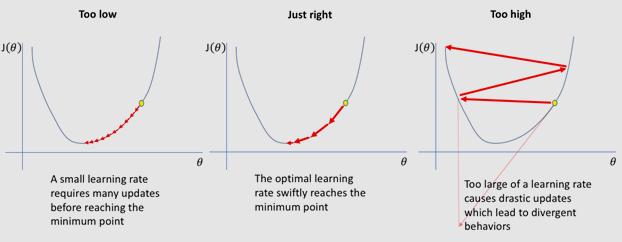
Công thức cập nhất GD:

= - α



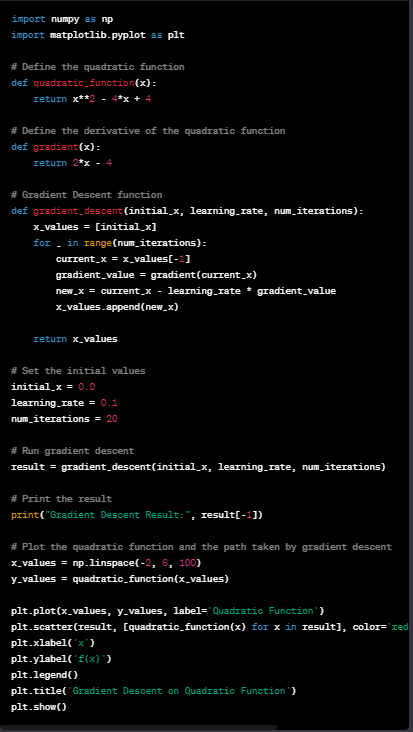
1.1.1.1 Gradient Descent 1

*Learning rate -* α*:* là một siêu tham số (Hyperparameter) quan trọng ảnh hưởng đến tốc độ học và độ chính xác trong huấn luyện mô hình, vô cùng quan trọng trong Gradient Descent. Nếu Learning Rate quá nhỏ thì thuật toán cần nhiều bước để hội tụ sẽ gây mất thời gian nhiều hơn. Ngược lại, Learning Rate quá lớn sẽ xảy ra sai sót, bỏ lỡ cực tiểu, và vượt ra ngoài khiến thuật toán không tìm được cực tiểu không thể hội tụ được.

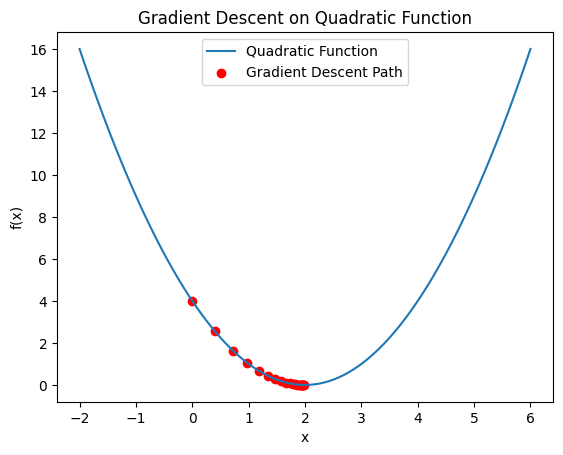


1.1.1.2 Learning Rate affect 1

Triển khai:



1.1.1.3 Gradient Descent Code Example 1



1.1.1.4 Kết quả Gradient Descent 1

**Multivariate Gradient Descent**

Hầu hết trong các tập dữ liệu của chúng đều có rất nhiều thuộc tính (variables/columns/dimensions) chứ không chỉ duy nhất một. Khi đó ta có công thức:

(,…,) = + + + … +

Về mặt tổng quan, các bước triển khai khi làm việc với tập dữ liệu chứa nhiều thuộc tính có tính tương đồng với khi xử lý một thuộc tính duy nhất. Tuy nhiên cần sửa đổi một số điều trước khi tiến hành thực thi:

1. Hàm mất mát (Loss Function)

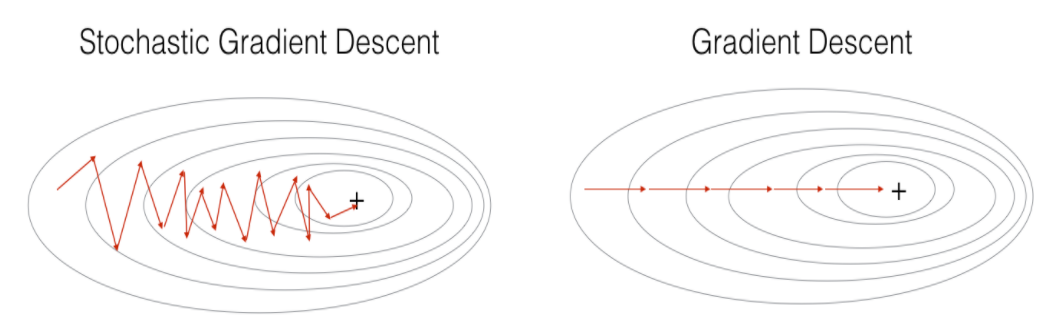
Loss = L = J =

* + 1. Stochastic Gradient Descent (SGD)

Stochastics Gradient Descent (SGD) lấy ngẫu nhiên một phần tử trong tập huấn luyện và tính lại vector độ dốc dựa chỉ dựa trên một điểm dữ liệu. Sau đó, nó lặp đi lặp lại quy trình này cho đến khi kết thúc, thay vì sử dụng toàn bộ tập huấn luyện. Ngoài ra, vì có rất ít dữ liệu cần xử lý ở mỗi vòng lặp, tính toán dựa trên một điểm dữ liệu sẽ khiến thuật toán chạy nhanh hơn. Điều này cũng giúp mô hình có thể được huấn luyện với dữ liệu lớn hơn vì mỗi vòng lặp chỉ cần điền một điểm dữ liệu vào bộ nhớ.

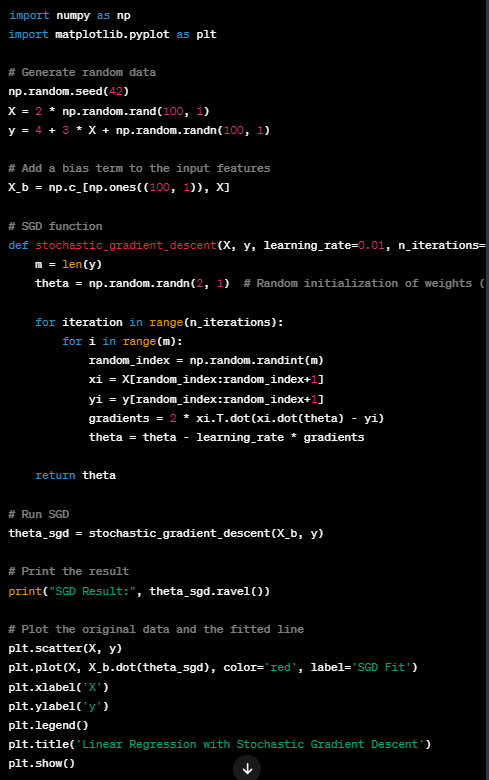
Mặt khác, do tính chất ngẫu nhiên của dữ liệu đưa vào trong quá trình huấn luyện, hàm chi phí của SGD sẽ tăng lên và giảm xuống theo thời gian thay vì giảm từ từ giống như Batch GD. Nghiệm của bài toán sẽ tiệm cận rất gần cực tiểu. Tuy nhiên, khi nó đến cực tiểu, giá trị hàm chi phí sẽ không ổn định. Khi điều kiện dừng, bộ tham số cuối cùng của chúng ta sẽ đủ tốt, nhưng không phải là tối ưu.

Thuật toán có thể thoát khỏi cực tiểu địa phương khi hàm chi phí liên tục thay đổi. Do đó, SGD có khả năng tìm được cực trị toàn cục hơn là chuyển động gradient đơn lẻ. Vì vậy, mặc dù lựa chọn ngẫu nhiên dữ liệu sẽ giúp thuật toán tránh nghiệm tối ưu cục bộ, nhưng nó cũng sẽ tránh nghiệm cực tiểu.

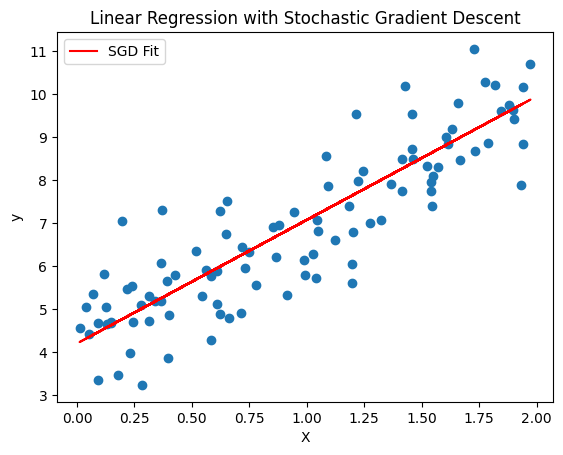


1.1.2.1 So sánh SGD và GD 1

Triển khai:



1.1.2.2 SGD code mẫu 1



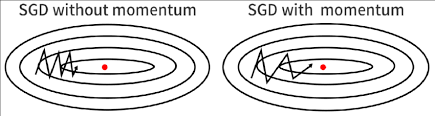
1.1.2.3 Kết quả thu được từ SGD 1

* + 1. Momentum

Quá trình đào tạo mạng lưới thần kinh được đẩy nhanh bằng cách sử dụng Momentum trong học máy và học sâu. Trong phương pháp này, một phần cập nhật trọng số trước đó được tích hợp vào bản cập nhật hiện tại trong quá trình tối ưu hóa.

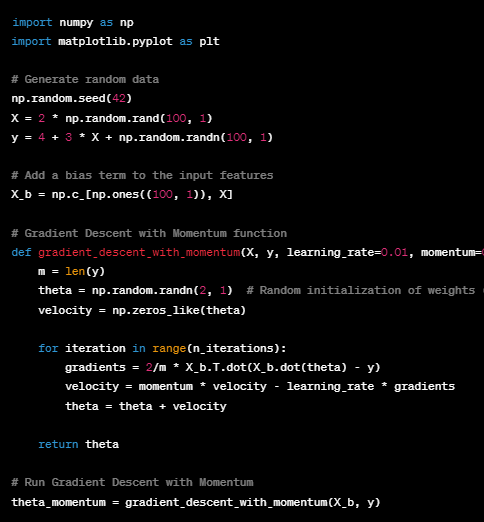
Độ dốc của hàm chi phí được tính cho mỗi trọng số trong mạng nơ-ron để tối ưu hóa động lượng. Một biến mới, số hạng động lượng, được thêm vào thay vì điều chỉnh trọng số dựa trên gradient. Theo cách tương tự như một đường trung bình động của các độ dốc, thuật ngữ này tích lũy các độ dốc trong quá khứ để hướng dẫn.

Khi bối cảnh tối ưu hóa nhiễu hoặc độ dốc thay đổi nhanh chóng, tối ưu hóa động lượng có vẻ hữu ích. Hiệu ứng làm mịn của nó đối với quá trình tối ưu hóa sẽ giúp nó tránh bị mắc kẹt ở mức cực tiểu cục bộ. Động lượng nhanh chóng đẩy nhanh quá trình đào tạo mạng lưới thần kinh sâu và cải thiện hiệu suất.

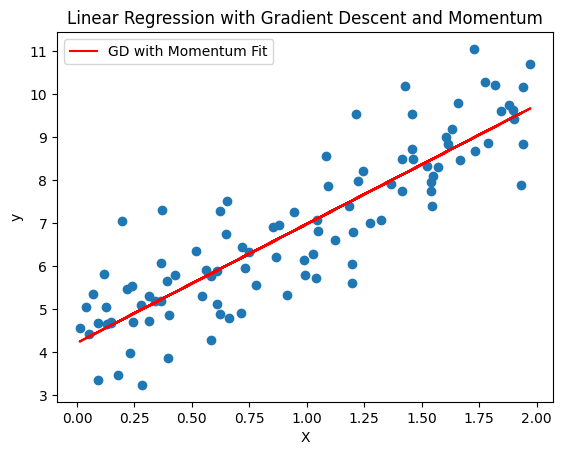


1.1.3.1 Momentum 1

Triển khai:



1.1.3.3 Momentum code 1



1.1.3.3 Kết quả Momentum 1

* + 1. Adagrad

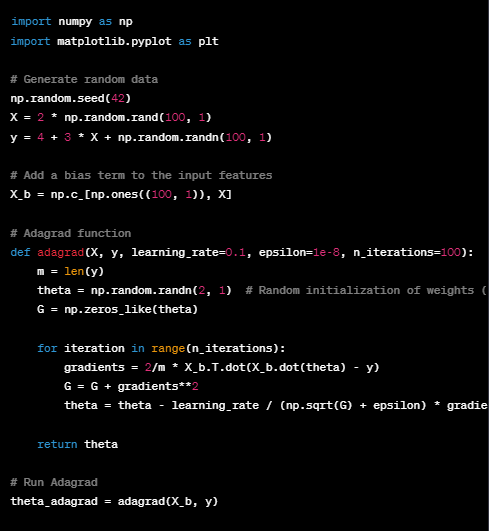
Adagrad cũng là một thuật toán tối ưu hóa dựa trên gradient descent. Thuật toán này hỗ trợ điều chỉnh tốc độ học ứng với nhiều tính năng khác nhau bằng cách cung cấp các cập nhật nhỏ hơn cho các tham số liên quan đến các tính năng thường xuyên xuất hiện và các cập nhật lớn h Vì lý do này, Adagrad rất phù hợp với việc xử lý dữ liệu nhỏ.Adaptive gradient, còn được gọi là Adagrad, là một thuật toán tối ưu hóa trong lĩnh vực học máy và học sâu. Công việc chính của nó là tối ưu hóa quá trình đào tạo mạng lưới thần kinh bằng cách điều chỉnh tốc độ học tập cho từng tham số.

Khả năng thích ứng của thuật toán Adagrad được đạt được bằng cách tăng tốc độ học cho từng tham số dựa trên độ dốc lịch sử của chúng. Các tham số có độ dốc cao có tốc độ học giảm, trong khi các tham số có độ dốc nhỏ có tốc độ học cao hơn. Cách tiếp cận sắc thái này thúc đẩy sự hội tụ tăng tốc trong quá trình đào tạo và ngăn chặn sự suy giảm nhanh chóng về tốc độ học tập đối với các tham số thường gặp.

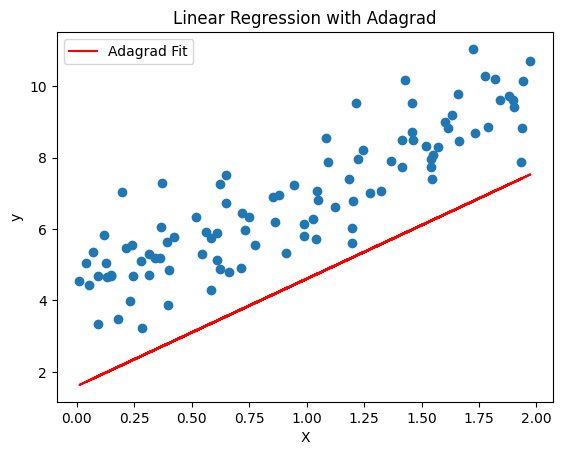
Adagrad thành công trong các trường hợp dữ liệu thưa thớt và không thường xuyên hoặc thiếu các tính năng đầu vào. Khả năng điều chỉnh tốc độ học cho từng tham số của nó chứng tỏ có lợi trong việc xử lý dữ liệu thưa thớt, góp phần cải thiện hiệu suất chung trong đào tạo mạng lưới thần kinh.

IMG_256

IMG_256



1.1.4.1 Adagrad code mẫu 1



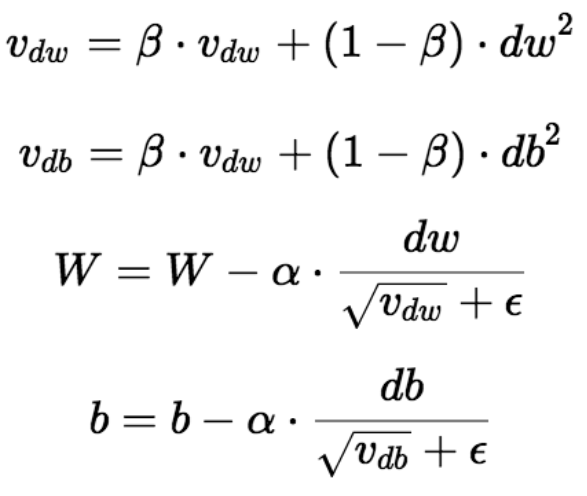
1.1.4.2 Adagrad Kết quả 1

* + 1. RMSprop

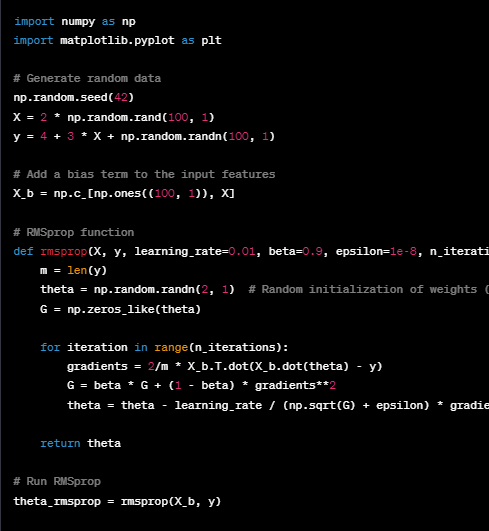
Một thuật toán tối ưu hóa quan trọng, RMSProp (viết tắt của Root Mean Square Propagation), được sử dụng trong học máy và học sâu để nâng cao việc đào tạo mạng lưới thần kinh.

Khác với Adagrad và Adadelta, RMSProp tự động điều chỉnh tốc độ học tập cho mỗi tham số trong quá trình đào tạo. RMSProp tính toán mức động trung bình của các gradient bình phương thay vì tích lũy tất cả các gradient trước đây như Adagrad. Phương pháp này ngăn chặn tình trạng giảm nhanh chóng bằng cách điều chỉnh tốc độ học tập dần dần.

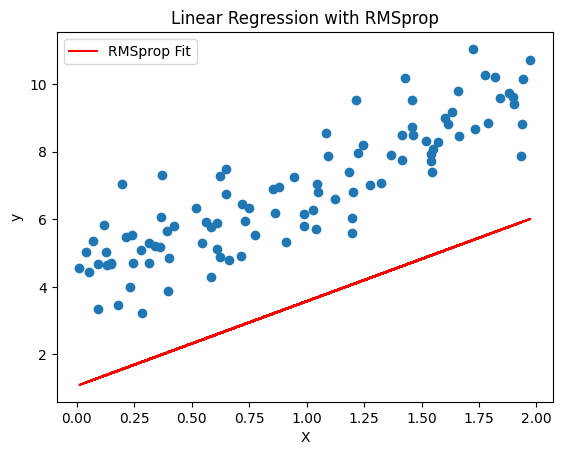
Ưu điểm chính của RMSProp là nó có thể xử lý các mục tiêu không cố định, chức năng cơ bản của mạng lưới thần kinh là ước tính những thay đổi theo thời gian. RMSProp điều chỉnh tốc độ học theo hàm mục tiêu đang phát triển, không giống như Adagrad, có thể hội tụ quá nhanh trong những tình huống như vậy. Khi hệ số phân rã được thêm vào, ảnh hưởng của các gradient trước đó sẽ được cải thiện. Các gradient gần đây sẽ nhận được nhiều trọng số hơn và các gradient cũ sẽ nhận được ít trọng số hơn.



Triển khai:



1.1.5.1 RMSProp code mẫu 1



1.1.5.2 Kết quả RMSprop 1

* + 1. Adam

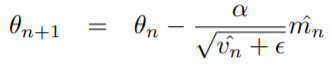
Adam, còn được gọi là Adaptive Moment Estimation, là một thuật toán tối ưu hóa gradient descent thường được sử dụng để huấn luyện mô hình máy học và học sâu. Nó kết hợp các ưu điểm của RMSprop và Momentum. Adam được thiết kế để cải thiện tốc độ học và hiệu suất của mô hình bằng cách giảm độ nhạy của learning rate với gradient.

Thuật toán Adam được xây dựng dựa trên hai thành phần chính: momentum-like term và squared gradient term.

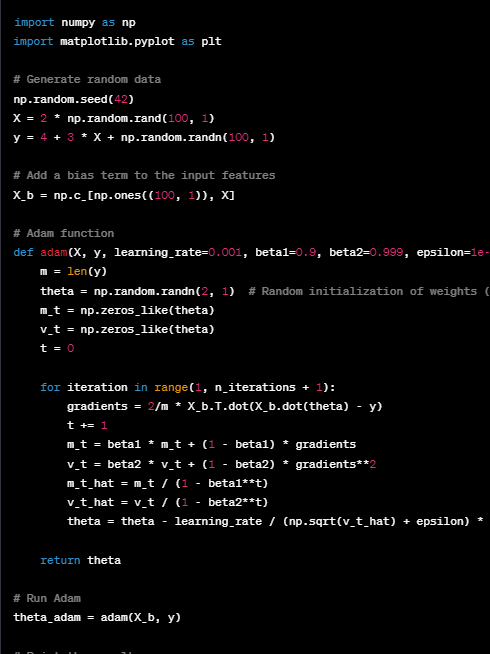
Adam, viết tắt của Ước tính thời điểm thích ứng, là một thuật toán tối ưu hóa hiệu quả cao được sử dụng trong học máy và học sâu để tối ưu hóa việc đào tạo mạng lưới thần kinh.

Adam đã kết hợp RMSProp và nguyên lý động lượng . Nó theo dõi giá trị trung bình và phương sai tương ứng trên đường trung bình động trong khoảnh khắc đầu tiên của gradient. Ngay cả khi độ dốc giảm dần, đường trung bình động của thời điểm đầu tiên, giống như động lượng, vẫn duy trì hướng của trình tối ưu hóa. Đồng thời, đường trung bình động của thời điểm thứ hai, giống như RMSProp, cho phép chia tỷ lệ tốc độ học thích ứng dựa trên phương sai độ dốc.

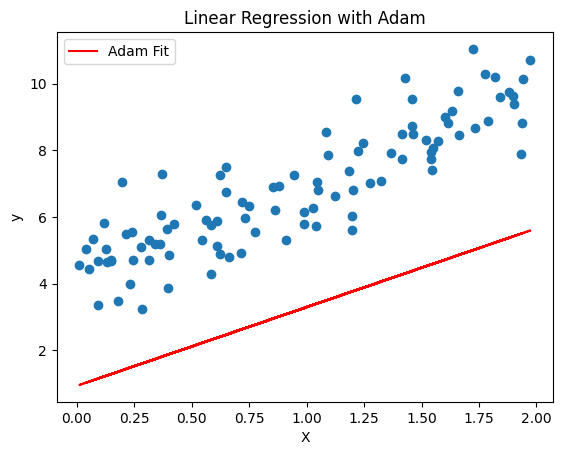
Adam kết hợp một bước điều chỉnh độ lệch quan trọng để điều chỉnh độ lệch về 0 trong giai đoạn tối ưu hóa ban đầu. Sự điều chỉnh này nâng cao hiệu suất của thuật toán trong quá trình đào tạo sớm. Đáng chú ý, Adam nổi bật nhờ khả năng tự chủ trong việc điều chỉnh siêu tham số, loại bỏ nhu cầu điều chỉnh thủ công các tham số như suy giảm tốc độ học tập hoặc hệ số xung lượng. Tính năng này góp phần vào sự phổ biến rộng rãi của nó.



Triển khai:



1.1.6.1 Adam code mẫu

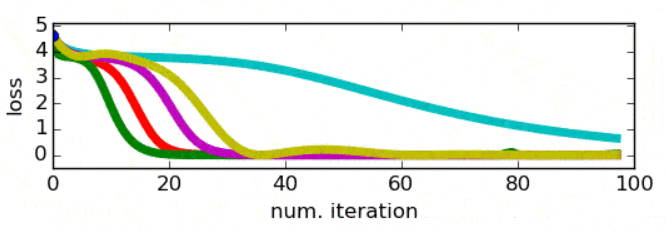


1.1.6.2 Kết quả Adam 1

* + 1. Tổng kết

|  |  |  |
| --- | --- | --- |
|  | Ưu điểm | Nhược điểm |
| Gradient Descent (GD) | - cơ bản, dễ hiểu  - giải quyết được vấn đề tối ưu model neural network bằng cách cập nhật trọng số sau mỗi vòng lặp | - Dễ bị lỗi ở local minimum  - Mất nhiều thời gian với tập dữ liệu lớn.  - Yêu cầu bộ nhớ lớn |
| Stochastic Gradient Descent (SGD) | - Khắc phục GD, làm việc được với dữ liệu lớn | - Vẫn còn phụ thuộc Learning rate và initial |
| Momentum | - Không bị lỗi local minimum giống GD mà vẫn có thể tiến tới Global minimum | - Có thể đi quá xa nếu không được đặt đúng siêu tham số. |
| Adagrad | -Learning rate sẽ được điều chỉnh tự động | - Tính toán tốn kém, đạo hàm bậc 2  - Learning rate sẽ giảm dần, dễ khiến đóng băng |
| RMSprop | - Giải quyết được việc giảm dần của Adagrad | - Kết quả nhận được có thể là local minimum |
| Adam | - Nhanh  - Khắc phục việc bị đóng băng, variance cao | - Tốn nhiều tài nguyên |

1.1.7.1 Bảng so sánh các Optimizers



1.1.7.1 So sánh các Optimizers 1

CHƯƠNG 2 – Continual Learning và Test Production khi xây dựng giải pháp học máy

* 1. Continual Learning

Học suốt đời (Lifelong ML - Continual Learning) có ba đặc điểm chính: Quá trình học liên tục, duy trì; tích lũy tri thức và sử dụng kiến thức đã học trong quá khứ để hỗ trợ nhiệm vụ mới. Ngoài ra nó cũng có thể phát hiện ra những nhiệm vụ mới và học chúng từng bước để thêm tri thức giúp cải thiện mô hình. Có một số mô hình học máy (ML) có các đặc trưng liên quan.

1. **Học chuyển giao (TRANSFER LEARNING)**

Học chuyển đổi, còn được gọi là học chuyển đổi hoặc học truyền dẫn, rất phổ biến trong khai phá dữ liệu và học máy. Trong xử lý ngôn ngữ tự nhiên, nó còn được gọi là thích ứng miền. Một miền nguồn và một miền đích là hai miền thông thường của học chuyển đổi. Mặc dù có thể có nhiều miền nguồn khác nhau, nhưng phần lớn các nghiên cứu hiện tại chỉ sử dụng một miền nguồn

Một lượng lớn dữ liệu huấn luyện thường có trong miền nguồn.

Miền đích không chứa dữ liệu huấn luyện được gắn nhãn.

Mục tiêu của học chuyển giao là sử dụng dữ liệu có gắn nhãn ở miền nguồn để giúp học trong miền đích. Từ miền nguồn sang miền đích có thể chuyển nhiều loại tri thức để giúp học trong miền đích..

1. **Học đa nhiệm vụ (MULTI-TASK LEARNING)**

Học đa nhiệm (MTL) là việc học nhiều nhiệm vụ cùng một lúc. Điều này cho phép họ sử dụng thông tin liên quan mà nhiều nhiệm vụ mang lại. Ngoài ra, nó giúp tổng quan tốt hơn vì nó ngăn chặn các nhiệm vụ cá nhân quá tải. Bây giờ chúng ta sẽ xem xét học đa nhiệm vụ theo lô, còn được gọi là học đa nhiệm vụ.

Định nghĩa: Học đa nhiệm vụ (MTL) liên quan đến việc học nhiều nhiệm vụ đồng thời với T={1,2,..,N}. Mỗi nhiệm vụ t thuộc T có Dt là dữ liệu đào tạo. Mỗi nhiệm vụ có mục tiêu tối đa hóa hiệu suất.

Học đa nhiệm có giám sát là chủ đề của phần lớn các công trình nghiên cứu hiện tại.

So sánh giữa học đa nhiệm và học suốt đời

Giống nhau:

Cả hai đều nhằm mục đích sử dụng một số thông tin được chia sẻ giữa các nhiệm vụ để giúp học.

Khác nhau:

Mô hình truyền thống liên quan đến học đa nhiệm, nhưng nó tối ưu hóa nhiều nhiệm vụ cùng một lúc thay vì chỉ tối ưu hóa một nhiệm vụ. Những đặc điểm của LL là nó không tích lũy kiến thức theo thời gian và nó không có khái niệm học liên tục. Vì vậy, việc giữ lại kiến thức là cần thiết để có thể học nhiều nhiệm vụ hơn với sự trợ giúp của kiến thức đã học từ các nhiệm vụ trước đó. Do đó, chúng tôi coi MTL trực tuyến hoặc gia tăng là LL.

1. **Học trực tuyến (ONLINE LEARNING)**

Trong tư cách là một mô hình học, các điểm dữ liệu huấn luyện đến theo thứ tự tuần tự. Mô hình hiện tại được cập nhật nhanh chóng để tạo ra mô hình tốt nhất. Nó thường được sử dụng khi tính toán không thể thực hiện được trên toàn bộ tập dữ liệu hoặc các ứng dụng thực tế không thể đợi cho đến khi thu thập được một lượng lớn dữ liệu dữ liệu huấn luyện. Điều này trái ngược với học hàng loạt cổ điển, nơi tất cả dữ liệu huẩn luyện có sẵn ngay từ đầu để đào tạo.

Sự khác biệt học trực tuyến và học suốt đời.

Mục đích của giao dịch rất khác với LL mặc dù họ học trực tuyến với dữ liệu trong tương lai trong luồn hoặc theo thứ tự tuần tự. Học trực tuyến vẫn thực hiện các nhiệm vụ học theo thời gian tương tự như trước đây. Mục tiêu của nó là nâng cao khả năng học tập bằng cách sử dụng dữ liệu đến từng bước. Ngoài ra, LL nhằm học từ nhiều nhiệm vụ khác nhau để giữ lại và sử dụng tri thức đã học để giúp học công việc trong tương lai.

1. **Học tăng cường (REINFORCEMENT LEARNING)**

Là một tác nhân học bằng cách thử nghiệm và tương tác với môi trường năng động Trạng thái hiện tại của môi trường được tác nhân nhận đầu vào trong mỗi bước tương tác. Chọn một trong số các hành động có thể. Môi trường bị ảnh hưởng bởi hành động này. Giá trị của quá trình chuyển đổi trạng thái này sẽ được đại lý nhận được, có thể là phần thưởng hoặc hình phạt. Khi tác nhân biết quỹ đạo của các hành động để tối ưu hóa mục tiêu của nó, quy trình này lặp đi lặp lại. Mục tiêu của học tăng cường là học chính

Sự khác biệt giữa học tăng cường và học suốt đời

Học tăng cường là học bằng cách thử và sai trong môi trường của nó, mang lại phản hồi hoặc phần thưởng cho đại lý, nhưng phản hồi này được giới hạn trong một nhiệm vụ và môi trường. Ý tưởng tích lũy tri thức không hữu ích cho công việc học trong tương lai.

1. **Siêu học (META-LEARNING)**

Mục tiêu là học một nhiệm vụ mới bằng cách sử dụng một mô hình đã được đào tạo về nhiều miền nhiệm vụ rất giống nhau. Nó thường được sử dụng để giải quyết vấn đề học một lần hoặc ít. Hệ thống siêu học thường bao gồm hai phần: bộ học cơ sở (hoặc bộ học nhanh) và bộ học siêu học (hoặc bộ học châm).

Bộ học cơ sở được đào tạo để hoàn thành nhiệm vụ trong một thời gian ngắn.

Người siêu học làm việc trong một không gian meta nhiệm vụ với mục đích chia sẻ kiến thức. Mô hình siêu học cho phép bộ học cơ sở học hiệu quả với rất ít ví dụ.

Sự khác biệt giữa siêu học và học suốt đời

Một mô hình meta được đào tạo bằng cách đào tạo từ một số lượng lớn các nhiệm vụ và nhanh chóng thích ứng với một nhiệm vụ mới chỉ với một số ví dụ. Một điều quan trọng khiến meta-learning bị giới hạn là nhiệm vụ đào tạo và kiểm tra/nhiệm vụ mới được tạo ra từ cùng một phân phối. Chúng tôi hy vọng rằng nhiều nhiệm vụ mới không giống như các nhiệm vụ cũ trong phần lớn các trường hợp thực tế. Các nhiệm vụ trong quá trình đánh giá thuật toán meta-learning

* 1. Test Production

Triển khai là quá trình mà một mô hình học máy phải trải qua để được sử dụng trong thế giới thực. Mô hình được tích hợp vào hệ thống sản xuất trong quá trình triển khai, điều này cho phép nó dự đoán hoặc đưa ra quyết định dựa trên dữ liệu mới, chưa từng được thấy trước đó. Để đảm bảo độ chính xác, độ tin cậy và hiệu suất của mô hình trong quá trình sản xuất, việc kiểm thử cũng rất quan trọng.

**1. Mở rộng mô hình:**

- Tích hợp: Việc tích hợp mô hình vào môi trường sản xuất thường đòi hỏi phải làm việc cùng với các kỹ sư phần mềm và đội ngũ công nghệ thông tin.

- Khả năng mở rộng: Đảm bảo rằng mô hình triển khai có khả năng xử lý lượng yêu cầu dự đoán dự kiến và có khả năng mở rộng để đáp ứng nhu cầu tăng lên.

**2. Kiểm tra thử nghiệm trong quá trình sản xuất:**

- A/B Testing: Thực hiện kiểm thử A/B để đánh giá hiệu suất của mô hình mới so với mô hình khác hoặc cơ sở. Điều này tạo điều kiện thuận lợi cho việc đưa ra quyết định dựa trên dữ liệu về hiệu suất của mô hình.

-Giám Sát Liên Tục: Thực hiện giám sát liên tục của mô hình trong môi trường sản xuất để phát hiện bất kỳ bất thường, sự thay đổi, hoặc giảm hiệu suất nào theo thời gian.

**3. Cam kết chất lượng:**

- Chất lượng dữ liệu: Xác minh rằng các giả định đã được thực hiện trong quá trình đào tạo mô hình không phù hợp với dữ liệu được sử dụng trong quá trình sản xuất. Hiệu suất của mô hình có thể bị ảnh hưởng bởi dữ liệu không nhất quán hoặc kém chất lượng.

- Chỉ số Hiệu suất: Tạo các chỉ số hiệu suất phù hợp với mục đích sử dụng cụ thể và sử dụng chúng để đánh giá hiệu suất của mô hình một cách đều đặn.

**4.Bảo mật:**

- Bảo mật dữ liệu: Thực hiện các biện pháp bảo mật để bảo vệ dữ liệu nhạy cảm được sử dụng bởi mô hình trong quá trình sản xuất, đảm bảo tuân thủ các quy định về quyền riêng tư và bảo mật.

- Bảo mật mô hình: Bảo vệ mô hình chính nó để ngăn người dùng không phép truy cập, sửa

đổi hoặc tấn công.

**5. Xử lý sai lầm:**

- Cơ chế đặc chế, còn được gọi là Fallback Mechanism, được sử dụng để xử lý các tình huống khi mô hình gặp phải các đầu vào không mong muốn hoặc lỗi.

- Ghi lỗi: Thiết lập một hệ thống ghi lỗi chi tiết để ghi lại và phân tích các vấn đề phát sinh trong quá trình sản xuất.

**6. Cải tiến thường xuyên:**

- Vòng Phản Hồi: Thiết lập vòng phản hồi để cải tiến liên tục. Quá trình cập nhật và đào tạo lại mô hình dựa trên thông tin từ hiệu suất mô hình và phản hồi của người dùng.

- Phiên bản mô hình: Hệ thống phiên bản được thiết lập để quản lý các cập nhật và thay đổi cho mô hình triển khai.

**7. Hồ sơ:**

- Tài liệu mô hình: Bảo trì tài liệu chi tiết liên quan đến kiến trúc, các phụ thuộc và quy trình triển khai của mô hình.

- Tài liệu người dùng: cung cấp cho người dùng cuối hoặc các bên liên quan thông tin liên quan đến tương tác, giải thích kết quả của mô hình.

**TÀI LIỆU THAM KHẢO**

1. [Gradient Descent, Quý Nguyễn](https://ndquy.github.io/posts/gradient-descent-2/" \l ":~:text=Gradient%20Descent%20là%20một%20thuật,cho%20việc%20tối%20ưu%20models.).
2. [Multivariable Regression and Gradient Descent. Rajkeshav](https://www.codingninjas.com/studio/library/multivariable-regression-and-gradient-descent).
3. [Gradient Descent, Anirudh Pai.](https://www.machinelearningworks.com/tutorials/gradient-descent)
4. [[Gradient Descent] – Phần 1: “Gradient Descent là gì?”, Khoa Nguyen Van.](http://tutorials.aiclub.cs.uit.edu.vn/index.php/2020/06/07/machine-learning-gradient-descent-la-gi-phan-1-5/)
5. [A Comprehensive Guide on Optimizers in Deep Learning, Ayush Gupta.](https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on-deep-learning-optimizers/" \l "Gradient_Descent_Deep_Learning_Optimizer)
6. [Optimizers in Deep Learning, Cathrine Jeeva.](https://www.scaler.com/topics/deep-learning/optimizers-in-deep-learning/)
7. [Various Optimization Algorithms For Training Neural Network, Sanket Doshi.](https://towardsdatascience.com/optimizers-for-training-neural-network-59450d71caf6)
8. [Optimizer- Hiểu sâu về các thuật toán tối ưu ( GD,SGD,Adam,..), Trần Trung Trực.](https://viblo.asia/p/optimizer-hieu-sau-ve-cac-thuat-toan-toi-uu-gdsgdadam-Qbq5QQ9E5D8)
9. [Types of Optimizers in Deep Learning Every AI Engineer Should Know, Pavan Vadapalli.](https://www.upgrad.com/blog/types-of-optimizers-in-deep-learning/)
10. [ĐÁNH GIÁ HIỆU NĂNG CỦA CÁC THUẬT TOÁN TỐI ƯUTRONG MÔ HÌNH HỌC SÂU ĐỐI VỚI BÀI TOÁN PHÂN LỚP HÌNH ẢNH, Khoa Tin học – Trường ĐHSP Huế (Vĩnh Anh Nghiêm Quân – Nguyễn Lê Trung Thành – Nguyễn Thị Lan Anh)](https://csdlkhoahoc.hueuni.edu.vn/data/2021/5/BaiDangHoiThao.pdf)
11. [ĐÁNH GIÁ CÁC THUẬT TOÁN TỐI ƯU ĐỐI VỚI MÔ HÌNH MẠNG NƠ-RONTÍCH CHẬP TRONG TÁC VỤ NHẬN DIỆN HÌNH ẢNH, Trường Đại học Khoa học, Đại học Huế (Vương Quang Phước, Nguyễn Đức Nhật Quang).](https://jos.husc.edu.vn/backup/upload/vol_18/no_1/668_fulltext_4.ĐTVT%20-%20Phuoc%20-%20Vuong%20Quang%20Phuoc.pdf)
12. [Chương 2. Mô hình hoc máy liên quan tới Lifelong Machine Learning, Tram Ho.](https://itzone.com.vn/vi/article/chuong-2-mo-hinh-hoc-may-lien-quan-toi-lifelong-machine-learning/)