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

NGUYỄN HOÀI NAM - 52000688

OPTIMIZER, CONTINUAL LEARNING, TEST PRODUCTION

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

NGUYỄN HOÀI NAM - 52000688

OPTIMIZER, CONTINUAL LEARNING, TEST PRODUCTION

BÁO CÁO CUỐI KÌ
NHẬP MÔN HỌC MÁY

Người hướng dẫn
PGS.TS. Lê Anh Cường

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023

LỜI CẢM ƠN

Em xin chân thành cảm ơn thầy Lê Anh Cường đã truyền đạt trực tiếp lại cho em rất nhiều kiến thức và kinh nghiệm, nhiệt tình hướng dẫn và giúp đỡ cho em trong suốt quá trình học. Thầy đã giúp em có thêm nhiều kiến thức về môn Nhập môn Học máy. Từ những tri thức mà thầy đã truyền tải, em đã hoàn thiện bài báo cáo cuối kỳ của môn Nhập môn Học máy.

Em cũng xin chân thành cảm ơn khoa Công Nghệ Thông Tin Trường Đại học Tôn Đức Thắng đã cho em cơ hội được làm bài báo cáo cuối kỳ. Trong quá trình làm bài báo cáo,em sẽ không tránh khỏi những sai sót, mong thầy có thể nhận xét và góp ý để nhóm em có thể nhận ra lỗi sai, những lỗ hổng kiến thức và đặc biệt là sự khắc phục để có thể áp dụng kiến thức cho quá trình học hỏi được những kinh nghiệm mới và có thể hoàn thiện bản thân cho những bài báo cáo tiếp theo và cả quá trình làm việc sau này.

Lời cuối cùng, em xin kính chúc thầy cô nhiều sức khỏe, hạnh phúc và luôn công tác tốt.

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

Tác giả

(Ký tên và ghi rõ họ tên)

Nam

Nguyễn Hoài Nam

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 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 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 23 tháng 12 năm 2023

Tác giả

(Ký tên và ghi rõ họ tên)

Nam

Nguyễn Hoài Nam

MỤC LỤC

DANH MỤC BẢNG BIỂU

DANH MỤC CÁC CHỮ VIẾT TẮT

GD	Gradient Descent
SGD	Stochastic Gradient Descent

CHƯƠNG 1. OPTIMIZER

1.1 Gradient Descent (GD)

1.1.1 Định nghĩa

Gradient Descent là một phương pháp trong thuật toán tối ưu được sử dụng để tìm giá trị nhỏ nhất của một hàm mất mát thông qua việc điều chỉnh các tham số của mô hình. Cụ thể, GD tập trung vào việc cập nhật các tham số theo hướng ngược với độ dốc của hàm mất mát. Quy trình này giúp chúng ta di chuyển dần dần về phía giá trị tối thiểu của hàm mất mát. Thuật ngữ "gradient" đề cập đến vector gradient, tức là vector chứa độ dốc của hàm mất mát theo mỗi tham số. Gradient này cho biết hướng và độ lớn mà hàm mất mát đang tăng nhanh nhất.

Công thức:

$$xnew = xold – learningrate.gradient(x) \quad (1.1)$$

Trong đó

- xnew: giá trị mới của tham số cập nhật.
- xold: giá trị hiện tại của tham số.
- learning rate: một siêu tham số quan trọng

1.1.2 Ưu, nhược điểm

1.1.2.1 Ưu điểm

Thuật toán gradient descent cơ bản, dễ hiểu.

Thuật toán giải quyết được vấn đề tối ưu model neural network bằng cách cập nhật trọng số sau mỗi vòng lặp.

1.1.2.2 Nhược điểm

Còn phụ thuộc vào nghiệm khởi tạo ban đầu và learning rate

Tốc độ học quá lớn sẽ khiến cho thuật toán không hội tụ, quanh quẩn bên đích vì bước nhảy quá lớn; hoặc tốc độ học nhỏ ảnh hưởng đến tốc độ training.

1.2 Stochastic Gradient Descent (SGD)

1.2.1 Định nghĩa

Stochastic Gradient Descent là một thuật toán tối ưu hóa lặp, thường được sử dụng trong machine learning và deep learning. Đây là một biến thể của gradient descent thực hiện cập nhật các tham số mô hình dựa trên gradient của hàm mất mát được tính trên một tập con ngẫu nhiên của dữ liệu huấn luyện thay vì trên toàn bộ tập dữ liệu.

Ý tưởng cơ bản của SGD là lấy một tập con nhỏ ngẫu nhiên của dữ liệu huấn luyện, được gọi là mini-batch, và tính gradient của hàm mất mát đối với các tham số mô hình chỉ sử dụng tập con đó. Gradient này sau đó được sử dụng để cập nhật các tham số. Quá trình được lặp lại với một mini-batch ngẫu nhiên mới cho đến khi thuật toán hội tụ hoặc đạt đến một điều kiện dừng được xác định trước.

1.2.2 Ưu, nhược điểm

1.2.2.1 Ưu điểm

Hội tụ nhanh hơn và yêu cầu bộ nhớ ít hơn, nên sẽ giải quyết được đối với các bộ dữ liệu lớn.

Mạnh mẽ hơn đối với dữ liệu nhiễu và không ổn định và có thể thoát khỏi các điểm tối thiểu cục bộ.

1.2.2.2 Nhược điểm

Còn phụ thuộc vào nghiệm khởi tạo ban đầu và learning rate.

Cần nhiều vòng lặp hơn để hội tụ so với gradient descent và tốc độ học cần được điều chỉnh cẩn thận để đảm bảo hội tụ.

1.3 Momentum

1.3.1 Định nghĩa

Momentum là một kỹ thuật tối ưu hóa được sử dụng trong machine learning và deep learning để tăng tốc quá trình huấn luyện của các mạng neural. Nó dựa trên

ý tưởng thêm một phần của cập nhật trước đó vào cập nhật hiện tại của trọng số trong quá trình tối ưu hóa.

Ý tưởng cơ bản của momentum là tích lũy đà từ các bước trước đó và sử dụng nó để cập nhật tham số của mô hình. Trong quá trình học, thay vì chỉ dựa vào gradient hiện tại để di chuyển đến điểm tối ưu, momentum giữ lại một phần của đà trước đó và thêm vào đà mới được tính từ gradient hiện tại. Điều này giúp giảm độ chói lọi và dao động của quá trình cập nhật tham số, giúp mô hình hội tụ một cách ổn định hơn.

Công thức:

$$xnew = xold - (gama.v + learningrate.gradient) \quad (2.2)$$

Trong đó:

- xnew: tọa độ mới
- xold: tọa độ cũ
- gama: parameter , thường = 0.9
- learningrate: tốc độ học
- gradient: đạo hàm của hàm f

1.3.2 Ưu, nhược điểm

1.3.2.1 Ưu điểm

Tiến được tới điểm global minimum.

Hội tụ nhanh hơn đối với các vấn đề có điều kiện kém.

1.3.2.2 Nhược điểm

Tuy momentum giúp hòn bi vượt dốc tiến tới điểm đích, tuy nhiên khi tới gần đích, nó vẫn mất khá nhiều thời gian giao động qua lại trước khi dừng hẳn.

1.4 Adagrad

1.4.1 Định nghĩa

Adagrad (Adaptive Gradient) là một thuật toán tối ưu hóa được sử dụng trong machine learning và deep learning để tối ưu hóa quá trình huấn luyện của các mạng neural.

Ý tưởng chính của Adagrad là điều chỉnh tốc độ học của từng trọng số dựa trên tổng bình phương của độ dốc (gradient) của trọng số đó từ tất cả các mẫu dữ liệu đã được xem xét cho đến thời điểm đó.

Công thức:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{G_t + \epsilon}} \cdot g_t \tag{3.3}$$

Trong đó:

- n: hằng số
- g_t: gradient tại thời điểm t
- ϵ: hệ số tránh lỗi (chia cho mẫu bằng 0)
- G: là ma trận chéo mà mỗi phần tử trên đường chéo (i,i) là bình phương của đạo hàm vectơ tham số tại thời điểm t

1.4.2 Ưu, nhược điểm

1.4.2.1 Ưu điểm

Adagrad có khả năng điều chỉnh tốc độ học của mỗi tham số một cách linh hoạt, giúp xử lý tốt hơn với dữ liệu thưa thớt.

Hữu ích khi xử lý dữ liệu thưa thớt, nơi một số đặc trưng đầu vào có tần suất thấp hoặc thiếu.

1.4.2.2 Nhược điểm

Tổng bình phương biến thiên sẽ lớn dần theo thời gian cho đến khi nó làm tốc độ học cực kì nhỏ, làm việc training trở nên đóng băng.

1.5 Adadelta

1.5.1 Định nghĩa

Adadelta là một biến thể của Adagrad để khắc phục tình trạng giảm tốc độ học ở Adagrad. Thay vì lưu lại tất cả gradient như Adagrad, Adadelta giới hạn tích lũy gradient theo cửa sổ có kích thước w xác định. Bằng cách này, Adadelta vẫn tiếp tục học sau nhiều bước cập nhật.

1.5.2 Ưu, nhược điểm

1.5.2.1 Ưu điểm

Có thể điều chỉnh tốc độ học một cách linh hoạt hơn Adagrad.

Giúp giảm vấn đề về việc cần thiết phải điều chỉnh tốc độ học thủ công.

1.5.2.2 Nhược điểm

Không thể sử dụng learning rate.

Quá trình điều chỉnh tốc độ có thể dẫn đến sự hội tụ chậm.

1.6 RMSprop

1.6.1 Định nghĩa

Tương tự như Adagrad và Adadelta, RMSProp điều chỉnh tốc độ học của mỗi tham số trong quá trình huấn luyện. Tuy nhiên, thay vì tích lũy tất cả các đạo hàm trước đó như Adagrad, RMSProp tính một trung bình động của các đạo hàm bình phương. Điều này cho phép thuật toán điều chỉnh tốc độ học một cách mượt mà hơn và ngăn chặn tốc độ học giảm quá nhanh.

Thuật toán RMSProp cũng sử dụng một hệ số giảm để kiểm soát ảnh hưởng của các đạo hàm trước đó lên tốc độ học. Hệ số giảm này cho phép thuật toán đặt nhiều trọng số hơn cho các đạo hàm gần đây và ít trọng số hơn cho các đạo hàm cũ hơn.

Công thức:

$$E[g^2]_t = 0{,}9E[g^2]_{t-1} + 0{,}1g_t^2 \quad (4.4)$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t \quad (5.5)$$

1.6.2 Ưu, nhược điểm

1.6.2.1 Ưu điểm

RMSprop giải quyết được vấn đề tốc độ học giảm dần của Adagrad.

Xử lý được các mục tiêu không ổn định.

1.6.2.2 Nhược điểm

Thuật toán RMSprop có thể cho kết quả nghiệm chỉ là local minimum chứ không đạt được global minimum.

1.7 Adam

1.7.1 Định nghĩa

Adam là sự kết hợp các khái niệm của cả Momentum và RMSProp. Adam tính toán giá trị thích ứng (adaptive) của tốc độ học cho từng tham số dựa trên trung bình cộng của độ dốc (momentum) và trung bình bình phương của độ dốc gần đây

1.7.2 Ưu, nhược điểm

1.7.2.1 Ưu điểm

Hiệu quả và dễ triển khai.

Áp dụng được cho các bộ dữ liệu lớn và các mô hình có số chiều cao.

1.7.2.2 Nhược điểm

Yêu cầu điều chỉnh cẩn thận các siêu tham số

1.8 Adamax

1.8.1 Định nghĩa

Adamax là một biến thể của thuật toán tối ưu hóa Adam được sử dụng trong machine learning và deep learning để tối ưu hóa quá trình huấn luyện của các mạng neural.

Giống như Adam, Adamax cũng duy trì một trung bình động của moment đầu và moment thứ hai của gradient. Tuy nhiên, thay vì sử dụng moment thứ hai của gradient như trong Adam, Adamax sử dụng norm L-infinity của gradient. Điều này hữu ích trong các tình huống mà gradient rất thưa thớt hoặc có phương sai rất cao.

1.8.2 Ưu, nhược điểm

1.8.2.1 Ưu điểm

Việc sử dụng norm L-infinity trong Adamax giúp làm cho thuật toán này ổn định hơn so với Adam khi xử lý gradient thưa thớt.

Hội tụ nhanh hơn và yêu cầu bộ nhớ ít hơn.

1.8.2.2 Nhược điểm

Tính toán phức tạp

1.9 So sánh các phương pháp Optimizer

Bảng 1.1: So sánh các phương pháp Optimizer

Optimizer	Ưu điểm	Nhược điểm
Gradient Descent	Thuật toán gradient descent cơ bản, dễ hiểu. Thuật toán giải quyết được vấn đề tối ưu model neural network bằng cách cập nhật trọng số sau mỗi vòng lặp.	Còn phụ thuộc vào nghiệm khởi tạo ban đầu và learning rate
Stochastic Gradient Descent	Đơn giản trong việc triển khai và thực hiện tính toán	Còn phụ thuộc vào nghiệm khởi tạo ban đầu và learning rate

	Hiệu quả cho các bộ dữ liệu lớn với không gian đặc trưng có số chiều cao	
Momentum	Giảm độ dao động trong quá trình huấn luyện. Hội tụ nhanh hơn đối với các vấn đề có điều kiện kém.	Tăng độ phức tạp của bài toán
Adagrad	Learning rate thích ứng được với từng tham số Hiệu quả cho dữ liệu thưa thớt	Tích lũy các đạo hàm bình phương trong mẫu số có thể làm cho tốc độ học giảm quá nhanh. Có thể dừng quá trình học sớm
Adadelta	Có thể điều chỉnh learning rate linh hoạt hơn Adagrad Không có siêu tham số về learning rate	Quá trình điều chỉnh tốc độ học có thể quá tích cực, dẫn đến việc hội tụ chậm.
RMSprop	Tốc độ học thích ứng cho mỗi tham số giới hạn sự tích lũy của gradients. Hiệu quả đối với mục tiêu không ổn định.	Có thể có tốc độ hội tụ chậm trong một số tình huống
Adam	Hiệu quả và dễ triển khai một cách đơn giản. Áp dụng được cho các bộ dữ liệu lớn và các mô hình có số chiều cao. Khả năng tổng quát tốt.	Yêu cầu điều chỉnh cẩn thận các siêu tham số

Adamax	Mạnh mẽ hơn trong không gian có số chiều cao. Hoạt động tốt khi có độ nhiễu trong gradients	Tính toán phức tạp

CHƯƠNG 2. CONTINUAL LEARNING

2.1 Continual learning

2.1.1 Định nghĩa

Continual learning hay còn được gọi là Lifelong Machine Learning (Học máy suốt đời) hay Học suốt đời (Lifelong Learning) là một mô hình học máy tiên tiến, quá trình học được thực hiện liên tục, tích lũy tri thức đã học từ các bài toán trước đó và sử dụng các tri thức này hỗ trợ cho bài toán học trong tương lai.

Học máy suốt đời nhằm bắt chước quá trình và khả năng học của con người, tích luỹ và duy trì tri thức đã học được từ các bài toán trước và không ngừng sử dụng tri thức đó để học và giải quyết bài toán mới. Tuy nhiên nhiệm vụ học liên tục là một thách thức lâu dài đối với học máy và mạng nơron và sự phát triển của các hệ thống trí tuệ nhân tạo.

2.1.2 Ứng dụng trong xây dựng giải pháp học máy để giải quyết bài toán

2.1.2.1 Ứng dụng di động và IoT

Khi xây dựng một ứng dụng học máy trên thiết bị di động để phân loại hình ảnh và thu thập dữ liệu từ cảm biến trên thiết bị IoT, có thể dùng Continual Learning để khi người dùng chụp ảnh mới hoặc khi có dữ liệu mới từ cảm biến, mô hình sẽ được cập nhật mà không cần phải huấn luyện lại từ đầu. Điều này giúp mô hình nhanh chóng học từ dữ liệu mới và cải thiện khả năng phân loại của nó mà không gặp vấn đề quên thông tin từ dữ liệu cũ.

2.1.2.2 An ninh và phát hiện đối tượng độc hại

Khi xây dựng một hệ thống an ninh dựa trên học máy để phát hiện các đối tượng độc hại trong video giám sát, có thể dùng Continual Learning để khi có một đối tượng độc hại mới xuất hiện, mô hình sẽ được cập nhật để nhận diện đối tượng này mà không làm ảnh hưởng đến khả năng nhận diện của nó đối với các đối tượng

đã biết trước đó. Điều này giúp hệ thống an ninh duy trì khả năng phát hiện đối tượng mới mà không mất đi khả năng phát hiện các đối tượng đã biết.

2.1.2.3 Phát triển mô hình dung năng

Khi xây dựng một ứng dụng học máy để dự đoán doanh số bán hàng dựa trên dữ liệu tiêu dùng, có thể dùng Continual Learning để khi xuất hiện dữ liệu mới về thị trường hoặc xu hướng tiêu dùng mới, mô hình sẽ được cập nhật để học từ dữ liệu mới này mà không cần phải huấn luyện lại từ đầu. Điều này giúp mô hình dự đoán doanh số bán hàng trong tương lai dựa trên thông tin mới nhất từ thị trường.

2.1.2.4 Học máy trong dữ liệu y tế

Khi xây dựng một hệ thống học máy để dự đoán các biểu hiện của một bệnh dựa trên dữ liệu y tế, có thể dùng Continual Learning để khi có dữ liệu y tế mới từ các bệnh nhân, mô hình sẽ được cập nhật để học từ các triệu chứng mới mà không làm mất đi khả năng dự đoán triệu chứng cũ. Điều này giúp hệ thống theo dõi các biểu hiện mới của bệnh và cung cấp dự đoán chính xác hơn về tình trạng sức khỏe.

2.2 Test Production

2.2.1 Định nghĩa

Trong lĩnh vực công nghệ thông tin, kiểm thử học máy có sự khác biệt so với kiểm thử phần mềm truyền thống. Trong khi kiểm thử phần mềm tập trung vào kiểm tra logic được viết sẵn, kiểm thử ML lại chú trọng đến việc kiểm tra logic được học bởi mô hình

Kiểm thử ML có thể được chia thành hai thành phần chính: kiểm thử và đánh giá

Chúng ta quen thuộc với đánh giá học máy khi chúng ta huấn luyện mô hình được và sau đó đánh giá hiệu suất trên một tập dữ liệu xác thực chưa được sử dụng trong quá trình huấn luyện. Đánh giá này được thực hiện thông qua các chỉ số như độ chính xác, AUC ROC và các biểu đồ trực quan như đường cong precision-recall.

Mặt khác, kiểm thử ML liên quan đến việc kiểm tra hành vi của mô hình. Các kiểm thử trước huấn luyện (pre-train tests), có thể thực hiện mà không cần các tham số đã được huấn luyện, giúp xác minh tính đúng đắn của logic được viết ra. Ví dụ, kiểm tra xác suất phân loại có nằm trong khoảng từ 0 đến 1 hay không. Các kiểm thử sau huấn luyện (post-train tests) kiểm tra xem liệu logic được học có như mong đợi hay không. Ví dụ, đối với bộ dữ liệu Titanic, chúng ta mong đợi phụ nữ có xác suất sống sót cao hơn nam giới.

2.2.2 Ứng dụng trong xây dựng giải pháp học máy để giải quyết bài toán

2.2.2.1 Kiểm tra hiệu suất của giải pháp trên môi trường sản xuất

Hiệu suất là một trong những yếu tố quan trọng nhất của một giải pháp học máy. Một giải pháp có hiệu suất cao sẽ có thể xử lý dữ liệu nhanh chóng và chính xác, đáp ứng được nhu cầu của người dùng.

Test production giúp kiểm tra hiệu suất của giải pháp trên môi trường sản xuất bằng cách thực hiện các bài kiểm tra với các kịch bản thực tế. Các kịch bản này có thể mô phỏng các tình huống mà giải pháp sẽ phải đối mặt trong môi trường sản xuất.

Ví dụ, một giải pháp học máy được sử dụng để phân loại các email spam cần phải có hiệu suất cao để có thể phân loại được các email spam một cách kịp thời. Test production có thể được sử dụng để kiểm tra hiệu suất của giải pháp bằng cách thực hiện các bài kiểm tra với các lượng dữ liệu lớn.

2.2.2.2 Kiểm tra khả năng chịu lỗi của giải pháp

Một giải pháp học máy cần phải có khả năng chịu lỗi cao để có thể hoạt động ổn định trong môi trường sản xuất. Trong môi trường sản xuất, các lỗi có thể xảy ra do nhiều nguyên nhân khác nhau, chẳng hạn như lỗi phần cứng, lỗi phần mềm, hoặc lỗi dữ liệu.

Test production giúp kiểm tra khả năng chịu lỗi của giải pháp bằng cách thực hiện các bài kiểm tra với các kịch bản lỗi. Các kịch bản này có thể mô phỏng các lỗi mà giải pháp có thể gặp phải trong môi trường sản xuất.

Ví dụ, một giải pháp học máy được sử dụng để điều khiển xe tự lái cần phải có khả năng chịu lỗi cao để có thể tiếp tục điều khiển xe trong trường hợp xảy ra lỗi. Test production có thể được sử dụng để kiểm tra khả năng chịu lỗi của giải pháp bằng cách thực hiện các bài kiểm tra với các tình huống lái xe bất thường.

2.2.2.3 Kiểm tra bảo mật của giải pháp

Một giải pháp học máy cần phải được bảo mật để tránh bị hacker đánh cắp dữ liệu hoặc gây ra các thiệt hại khác.

Test production giúp kiểm tra khả năng bảo mật của giải pháp bằng cách thực hiện các bài kiểm tra với các kịch bản tấn công bảo mật. Các kịch bản này có thể mô phỏng các cuộc tấn công bảo mật mà giải pháp có thể gặp phải trong môi trường sản xuất.

Ví dụ, một giải pháp học máy được sử dụng để lưu trữ dữ liệu cá nhân cần phải được bảo mật để tránh bị hacker đánh cắp dữ liệu. Test production có thể được sử dụng để kiểm tra khả năng bảo mật của giải pháp bằng cách thực hiện cbàikiểm tra với các lỗ hổng bảo mật phổ biến.

TÀI LIỆU THAM KHẢO

Tiếng Việt

Trần Trung Trực. (2020). *Optimizer- Hiểu sâu về các thuật toán tối ưu (GD,SGD,Adam,..)*. Truy cập 23/12/2023. https://viblo .asia/p/optimizer-hieu-sau-ve-cac-thuat-toan-toi-uu-gdsgdadam-Qbq5QQ9E5D8

Tiếng Anh

amananandrai. (2023). 10 famous Machine Learning Optimizers. Retrieved December 23, 2023. https://dev.to/amananandrai/10-famous-machine-learning-optimizers-1e22#:~:text=Optimizers%20are%20algorithms%20used%20to,on%20a%20minimum%20loss%20value.

ApplyingML. (2023). Machine Learning in Production - Testing. Retrieved December 23, 2023. https://applyingml.com/resources/testing-ml/