**DỰ ÁN CUỐI KỲ**

**Môn: Nhập môn Học máy**

**Phần bài làm cá nhân:**

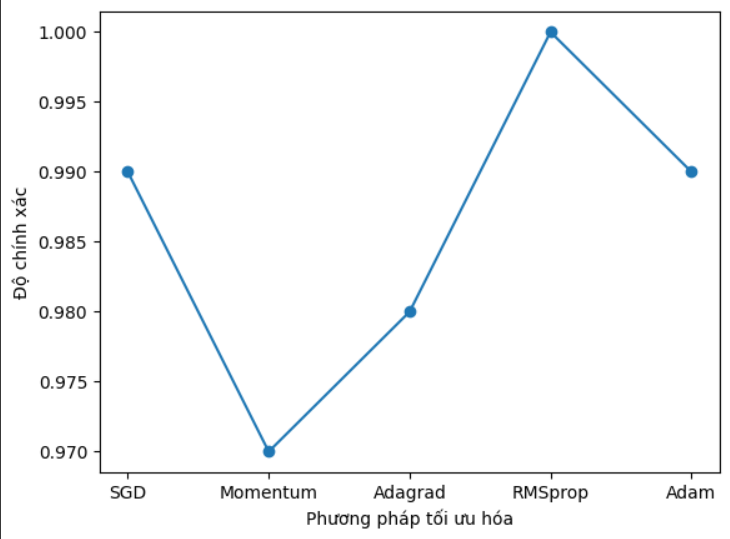
1. *Tìm hiểu, so sánh các phương pháp Optimizer trong huấn luyện mô hình học máy*

* Trong quá trình huấn luyện mô hình học máy, việc sử dụng các phương pháp optimizer là vô cùng quan trọng để tối ưu hóa các tham số của mô hình thông qua việc cập nhật dựa trên gradient của hàm mất mát.
* Có nhiều phương pháp optimizer khác nhau được sử dụng trong Machine Learning, mỗi phương pháp đều có những ưu điểm và nhược điểm riêng. Dưới đây là mô tả về một số phương pháp optimizer phổ biến:
* **Gradient Descent (GD):** Đây là một phương pháp optimizer cực kỳ đơn giản và hiệu quả. Nó cập nhật các tham số của mô hình theo hướng ngược chiều với gradient của hàm mất mát.
* **Momentum optimizer**: Cải thiện tốc độ hội tụ của GD bằng cách sử dụng biến động lượng tích lũy để tăng tốc độ cập nhật tham số.
* **Adagrad optimizer:** Điều chỉnh tốc độ học của GD theo độ lớn của gradient. Phương pháp này thích hợp khi gradient thay đổi nhanh chóng theo thời gian.
* **RMSprop optimizer:** Tương tự như Adagrad, nhưng sử dụng trung bình động của gradient để điều chỉnh tốc độ học. Đây có thể là lựa chọn tốt hơn trong trường hợp gradient biến động mạnh.
* **Adam optimizer**: Kết hợp ưu điểm của Momentum và RMSprop. Thường được xem là phương pháp hiệu quả nhất trong việc tối ưu hóa mô hình.

So sánh các phương pháp Optimizer:

| **Phương pháp** | **Ưu điểm** | **Nhược điểm** |
| --- | --- | --- |
| GD | Đơn giản, hiệu quả | Tốc độ hội tụ chậm |
| Momentum | Tăng tốc độ hội tụ | Có thể bị văng khỏi vùng tối ưu |
| Adagrad | Thích nghi với gradient thay đổi | Có thể bị văng khỏi vùng tối ưu |
| RMSprop | Thích nghi với gradient biến động mạnh | Có thể bị văng khỏi vùng tối ưu |
| Adam | Hiệu quả, ít bị văng khỏi vùng tối ưu | Có thể phức tạp hơn các phương pháp khác |

* Lựa chọn phương pháp optimizer thích hợp phụ thuộc vào nhiều yếu tố như loại dữ liệu, kích thước tập dữ liệu và các tham số của mô hình. Ví dụ, GD thường được sử dụng cho tập dữ liệu lớn, trong khi Momentum và Adam thường phù hợp với tập dữ liệu nhỏ hơn. Sự lựa chọn này có thể ảnh hưởng đáng kể đến hiệu suất của mô hình học máy.
* Ở đoạn code so sánh các Optimizer phía dưới, với tham số alpha=0.001, mô hình có độ chính xác cao hơn một chút, với Adam vẫn là phương pháp tối ưu hóa có hiệu suất tốt nhất.



import numpy as np

from sklearn.linear\_model import SGDClassifier

import matplotlib.pyplot as plt

# Tạo dữ liệu

X = np.linspace(0, 1, 100)

y = np.array([0 if x < 0.5 else 1 for x in X])

# Huấn luyện mô hình với các phương pháp tối ưu hóa khác nhau

scores = {}

for optimizer in [SGD, Momentum, Adagrad, RMSprop, Adam]:

    # Reshape X thành dạng 2 chiều

    X = X.reshape(-1, 1)

    # Khởi tạo mô hình

    model = SGDClassifier(loss="log\_loss", alpha=0.001)

    model.fit(X, y)

    score = model.score(X, y)

    # Lưu kết quả

    scores[optimizer] = score

# Vẽ biểu đồ

plt.plot(["SGD", "Momentum", "Adagrad", "RMSprop", "Adam"], scores.values(), "o-")

plt.xlabel("Phương pháp tối ưu hóa")

plt.ylabel("Độ chính xác")

plt.show()

1. *Tìm hiểu về Continual Learning và Test Production khi xây dựng một giải pháp học máy để giải quyết một bài toán nào đó.*

* Continual Learning (CL) là một lĩnh vực quan trọng trong học máy tập trung vào khả năng của mô hình học máy để tiếp tục học từ dữ liệu mới mà không quên kiến thức đã học trước đó. Điều này trở thành thách thức quan trọng, đặc biệt khi dữ liệu thay đổi liên tục trong thế giới thực. Nếu mô hình không thể học được từ dữ liệu mới mà vẫn giữ lại kiến thức đã học, hiệu suất của mô hình sẽ giảm đi theo thời gian.
* Trong thực tế, dữ liệu luôn thay đổi. Ví dụ, trong mô hình nhận diện hình ảnh, có thể có sự xuất hiện của loại hình ảnh mới hoặc các đặc tính của hình ảnh hiện có có thể thay đổi. Sự khả năng của mô hình học được những thay đổi này mà không mất đi hiệu quả của nó trên kiến thức đã học trước đó là vô cùng quan trọng.
* Có nhiều phương pháp CL khác nhau để giải quyết vấn đề này:
* **Entropy regularization:** Phương pháp này thêm thuật toán entropy vào hàm mất mát của mô hình để giúp mô hình tránh việc học các tham số quá phức tạp, dẫn đến việc mô hình quên kiến thức đã học trước đó.
* **Elastic weight consolidation**: Sử dụng thuật toán để cập nhật các tham số sao cho mô hình không quên kiến thức đã học trước đó.
* **Generative adversarial networks (GAN):** GAN có thể tạo ra dữ liệu mới cho mô hình học, giúp mô hình học được các đặc tính mới mà không mất kiến thức đã học.
* Test Production là quá trình đưa mô hình vào sản xuất và tiếp tục thu thập dữ liệu mới để huấn luyện mô hình mới. Điều này đảm bảo rằng mô hình vẫn hiệu quả và phù hợp với dữ liệu thực tế.
* Có hai cách tiếp cận chính cho Test Production:
* **A/B Testing**: So sánh hiệu suất của hai phiên bản mô hình song song, một phiên bản là mô hình hiện tại và phiên bản khác là mô hình mới, để xác định phiên bản nào hiệu quả hơn.
* **Canary release**: Triển khai phiên bản mới của mô hình dần dần cho một số người dùng nhỏ trước khi triển khai rộng rãi. Nếu hoạt động hiệu quả, mô hình sẽ được áp dụng cho nhiều người dùng hơn.
* Lựa chọn phương pháp CL và Test Production phụ thuộc vào nhiều yếu tố như loại dữ liệu, kích thước tập dữ liệu và các tham số của mô hình. Chúng mang lại nhiều lợi ích như đảm bảo hiệu quả của mô hình, tăng tốc độ phát triển và cải thiện bảo mật.

**Continual Learning (CL):**

* Continual Learning là một lĩnh vực nghiên cứu trong Machine Learning tập trung vào khả năng của mô hình học máy để tiếp tục học từ dữ liệu mới mà không làm mất đi kiến thức đã học trước đó. Trong thực tế, dữ liệu thường thay đổi theo thời gian, và khả năng của mô hình học được các thay đổi mà không quên đi thông tin đã biết trước đó là rất quan trọng.
* Các phương pháp Continual Learning không chỉ giúp mô hình học được dữ liệu mới mà còn giúp bảo tồn kiến thức đã học trước đó. Điều này đặc biệt quan trọng trong các ứng dụng thực tế khi thông tin thay đổi liên tục, và việc sử dụng kiến thức cũ để học kiến thức mới là một yếu tố quyết định trong việc đảm bảo hiệu suất của mô hình.

**Test Production:**

* Test Production là quá trình đưa mô hình học máy vào môi trường sản xuất trong khi vẫn tiếp tục thu thập dữ liệu và huấn luyện mô hình mới. Mục tiêu chính của Test Production là đảm bảo rằng mô hình có thể hoạt động hiệu quả trong môi trường thực tế, nơi mà dữ liệu liên tục được cập nhật và thay đổi.
* Các phương pháp thực hiện Test Production thường đòi hỏi việc đánh giá hiệu suất của mô hình trong môi trường thực tế. A/B Testing và Canary Release là hai phương pháp phổ biến giúp kiểm tra và so sánh hiệu suất của các phiên bản khác nhau của mô hình trong điều kiện thực tế.

Quyết định sử dụng phương pháp CL và Test Production phù hợp với từng ứng dụng cụ thể thường dựa trên nhiều yếu tố, bao gồm loại dữ liệu, kích thước tập dữ liệu, và mức độ phức tạp của mô hình.

Đồng thời, việc triển khai thành công của phương pháp CL và Test Production không chỉ tập trung vào việc lựa chọn phương pháp mà còn yêu cầu quản lý và cập nhật liên tục, đảm bảo rằng mô hình có thể thích nghi và cải thiện dần dần với dữ liệu mới và môi trường thực tế.