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

BÁO CÁO CUỐI KỲ MÔN NHẬP MÔN HỌC MÁY

DỰ ÁN CUỐI KỲ

Người hướng dẫn: **TS LÊ ANH CƯỜNG**

Người thực hiện: **LÊ TRẦN QUỲNH NHƯ – 52000379**

Lớp **: 20050401**

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

BÁO CÁO CUỐI KỲ MÔN NHẬP MÔN HỌC MÁY

DỰ ÁN CUỐI KỲ

Người hướng dẫn: **TS LÊ ANH CƯỜNG**
Người thực hiện: **LÊ TRẦN QUỲNH NHƯ – 52000379**
Lớp **: 20050401**
Khoá **: 24**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023

LỜI CẢM ƠN

Trong thời gian hoàn thành bài báo cáo vừa qua, em đã nhận được rất nhiều sự giúp đỡ, hướng dẫn và hỗ trợ tận tình từ quý thầy cô và các bạn. Em cảm thấy vô cùng biết ơn và muốn gửi những lời cảm ơn sâu sắc đến thầy cô, bạn bè và gia đình đã giúp bài báo cáo của em đạt kết quả tốt như hiện nay.

Đặc biệt, em muốn bày tỏ sự biết ơn sâu sắc và lòng kính trọng đến thầy Lê Anh Cường, thầy là người đã giảng dạy em môn Nhập môn Học máy trong suốt học kỳ vừa qua. Trong quá trình học tập, thầy đã truyền đạt cho em vô vàn kiến thức hay và bổ ích, giúp em có được cơ sở lý thuyết vững vàng để em vượt qua bài báo cáo này dễ dàng hơn.

Bài báo cáo này cũng không thể tránh khỏi những thiếu sót và hạn chế, em rất mong quý thầy cô sẽ bỏ qua cho em và chỉ bảo thêm để giúp em có điều kiện bổ sung và làm tốt hơn trong những bài báo cáo sau này.

Em xin kính chúc quý thầy, quý cô và quý nhà trường luôn mạnh khỏe, hạnh phúc và ngày một thành công hơn trong sự nghiệp trồng người của mình.

ĐỒ ÁN 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 luận văn 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 luận văn 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 20 tháng 12 năm 2023

Tác giả

(ký tên và ghi rõ họ tên)

Lê Trần Quỳnh Như

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)

TÓM TẮT

1. Vấn đề nghiên cứu:

Bài làm gồm có 2 phần:

- Phần 1: Bài 1 - Làm riêng từng người.
- Phần 2: Bài 2 - Làm chung trong nhóm.

2. Các hướng tiếp cận

- Lý thuyết.
- Thực hành.

3. Cách giải quyết vấn đề

Xem lại những nội dung đã học qua slide bài giảng, các kiến thức được ghi chép lại trong quá trình học và nghiên cứu thêm các video bài giảng trên mạng. Vận dụng chúng vào để giải quyết các nội dung trong đề thi.

4. Một số kết quả đạt được

Ôn lại được những kiến thức đã học, nắm vững các lý thuyết và phương pháp làm bài môn Nhập môn Học máy. Rèn luyện tư duy logic cho việc học tập các môn học sau.

MỤC LỤC

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

DANH MỤC BẢNG

DANH MỤC HÌNH

BÀI 1 - LÀM RIÊNG TỪNG NGƯỜI

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

1.1.1 Tìm hiểu các phương pháp Optimizer

1.1.1.1 Stochastic Gradient Descent (SGD)

- Khái niệm: SGD là một phương pháp tối ưu hóa dựa trên việc cập nhật trọng số của mô hình dựa trên gradient của hàm mất mát tính toán từ một mini-batch ngẫu nhiên trong dữ liệu huấn luyện. Việc sử dụng mini-batch giúp giảm độ phức tạp tính toán và tiết kiệm bộ nhớ so với việc sử dụng toàn bộ dữ liệu.
- Ưu điểm:
  - Dễ triển khai và tính toán hiệu quả.
  - Hiệu quả cho bộ dữ liệu lớn với không gian đặc trưng về chiều cao.
- Nhược điểm:
  - Có thể bị mắc kẹt trong các điểm tối ưu cục bộ.
  - Nhạy cảm cao đối với tốc độ học ban đầu.

1.1.1.2 Stochastic Gradient Descent với Gradient Clipping

- Khái niệm: SGD với Gradient Clipping làs một biến thể của SGD, nơi mà sau khi tính toán gradient cho mỗi mini-batch, gradient sẽ được cắt giữ nếu vượt qua một ngưỡng nhất định. Điều này giúp tránh tình trạng "gradient explosion" khi giá trị của gradient quá lớn.
- Ưu điểm:
  - Giảm khả năng gradient bùng nổ.
  - Cải thiện sự ổn định trong quá trình huấn luyện.
- Nhược điểm: Clipping có thể giúp giấu đi những vấn đề khác, như khởi đầu kém hoặc tỷ lệ học không tốt.

1.1.1.3 Momentum

- Khái niệm: Momentum là phương pháp tối ưu hóa giúp giảm độ dao động và tăng tốc quá trình học bằng cách tích luỹ trọng số của các bước trước đó. Nó có thể được xem như một quả cầu đang lăn xuống thung lũng, giúp mô hình vượt qua các "đỉnh núi" và đạt được đến điểm tối ưu.
- Ưu điểm:
  - Giảm độ dao động trong quá trình huấn luyện.
  - Hội tụ nhanh hơn đối với các vấn đề kém điều kiện.
- Nhược điểm: Tăng sức phức tạp của thuật toán.

1.1.1.4 Nesterov Momentum

- Khái niệm: Nesterov Momentum cải thiện ý tưởng của Momentum bằng cách đầu tiên cập nhật trọng số với một phần của quãng đường đã di chuyển trước đó, sau đó tính toán gradient tại vị trí mới. Điều này giúp dự đoán hướng di chuyển hiệu quả hơn và giảm nguy cơ overshooting.
- Ưu điểm:
  - Hội tụ nhanh hơn so với momentum cổ điển.
  - Giảm nguy cơ quá mức.
- Nhược điểm: Đòi hỏi chi phí tính toán cao hơn so với momentum klasik.

1.1.1.5 Adagrad

- Khái niệm: Adagrad là một phương pháp tối ưu hóa thích ứng tốc độ học cho mỗi tham số bằng cách sử dụng lịch sử của gradient. Cụ thể, nó điều chỉnh tốc độ học của mỗi tham số dựa trên tổng bình phương của gradient đã tính toán cho tham số đó.
- Ưu điểm:
  - Tốc độ học thích ứng cho mỗi tham số.
  - Hiệu quả cho dữ liệu thưa thớt.

- Nhược điểm:
  - Sự tích lũy gradient có thể làm giảm tốc độ học quá nhanh.
  - Có thể dừng học quá sớm.

1.1.1.6 Adadelta

- Khái niệm: Adadelta là biến thể của Adagrad, nhưng thay vì tích luỹ tất cả gradient, nó chỉ lưu một phần lịch sử giới hạn, giúp giảm bớt vấn đề về quá trình tích luỹ quá nhanh và giảm bớt yêu cầu bộ nhớ.
- Ưu điểm:
  - Có thể thích ứng với tốc độ học một cách động hơn so với Adagrad.
  - Không yêu cầu tham số tốc độ học.
- Nhược điểm: Tính ổn định của việc thích ứng tốc độ học có thể làm giảm tốc độ học quá chậm.

1.1.1.7 RMSProp

- Khái niệm: RMSProp là một biến thể khác của Adagrad, nơi mà nó giữ một trọng số động để giảm tác động của các gradient quá cũ và giúp tối ưu hóa mô hình trên các môi trường không ổn định.
- Ưu điểm:
  - Tốc độ học thích ứng cho mỗi tham số, giới hạn tích lũy gradient.
  - Hiệu quả cho các mục tiêu không ổn định.
- Nhược điểm: Có thể có tốc độ hội tụ chậm trong một số tình huống.

1.1.1.8 Adam

- Khái niệm: Adam (Adaptive Moment Estimation) kết hợp ý tưởng của Momentum và RMSProp. Nó duy trì một trung bình động của gradient và bình phương của gradient, giúp tối ưu hóa mô hình trên nhiều loại dữ liệu và kiến trúc mô hình.
- Ưu điểm:

  - Hiệu quả và dễ triển khai.
  - Áp dụng cho bộ dữ liệu lớn và mô hình có số chiều cao.
  - Khả năng tổng quát hóa tốt.
- Nhược điểm: Yêu cầu điều chỉnh cẩn thận của siêu tham số.

1.1.1.9 Adamax

- Khái niệm: Adamax là một biến thể của Adam, sử dụng norm L-infinity của gradient thay vì bình phương gradient để giảm bớt ảnh hưởng của gradient nhiễu và giảm yêu cầu bộ nhớ.
- Ưu điểm:
  - Ổn định hơn trong không gian có số chiều cao.
  - Hoạt động tốt khi có gradient nhiễu.
- Nhược điểm: Yêu cầu chi phí tính toán cao.

1.1.1.10 SMORMS3

- Khái niệm: SMORMS3 là một biến thể của RMSProp, nhưng thay vì lấy bình phương gradient, nó sử dụng bình phương của cube root của bình phương gradient. Điều này giúp tránh việc tốc độ học giảm quá nhanh và làm giảm khả năng bị rơi vào điểm tối ưu cục bộ.
- Ưu điểm:
  - Hiệu suất tốt trên bộ dữ liệu lớn với không gian chiều cao.
  - Ổn định trong môi trường có gradient nhiễu.
- Nhược điểm: Chi phí tính toán cao.

1.1.2 So sánh các phương pháp Optimizer

STT	Optimizer	Ưu điểm	Nhược điểm
1	Stochastic Gradient Descent (SGD)	- Dễ triển khai và hiệu quả về mặt tính toán. - Hiệu quả cho các bộ dữ liệu lớn với không gian đặc trưng về chiều cao.	- SGD có thể mắc kẹt trong các điểm tối thiểu cục bộ. - Rất nhạy cảm với tỷ lệ học ban đầu.
2	Stochastic Gradient Descent với Gradient Clipping	- Giảm khả năng gradient bùng nổ. - Cải thiện sự ổn định trong quá trình huấn luyện.	Clipping có thể giúp giấu đi những vấn đề khác, như khởi đầu kém hoặc tỷ lệ học không tốt.
3	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 thuật toán.
4	Nesterov Momentum	- Hội tụ nhanh hơn so với momentum cổ điển. - Có thể giảm thiểu hiện tượng vượt mục tiêu.	Đắt đỏ hơn so với momentum cổ điển.
5	Adagrad	- Tốc độ học thích ứng cho mỗi tham số. - Hiệu quả cho dữ liệu thưa thớt.	- Tích lũy gradient bình phương có thể làm giảm quá nhanh tốc độ học. - Có thể dừng học quá sớm.
	Adadelta	- Có thể điều chỉnh tốc độ	Quá trình điều chỉnh tốc

6		học một cách linh hoạt hơn cả Adagrad. - Không có siêu tham số tốc độ học.	độ học có thể quá mạnh mẽ, dẫn đến hội tụ chậm.
7	RMSProp	- Tốc độ học thích ứng cho mỗi tham số giới hạn sự tích lũy của gradient. - Hiệu quả cho các mục tiêu không ổn định.	Có thể có tốc độ hội tụ chậm trong một số tình huống.
8	Adam	- Hiệu quả và dễ triển khai. - Áp dụng cho các bộ dữ liệu lớn và mô hình chiều cao. - Khả năng tổng quát tốt.	Yêu cầu điều chỉnh cẩn thận của siêu tham số.
9	Adamax	- Ổn định hơn trong không gian chiều cao. - Hoạt động tốt khi có gradient nhiễu.	Tính toán đòi hỏi cao.
10	SMORMS3	- Hiệu suất tốt trên các bộ dữ liệu lớn với không gian chiều cao. - Ổn định trong điều kiện gradient nhiễu.	Tính toán đòi hỏi cao.

Bảng 1. 1 So sánh các phương pháp Optimizer

1.2 Tìm hiểu về Continual Learning và Test Production khi xây dưng một giải pháp học máy để giải quyết một bài toán nào đó

1.2.1 Continual Learning

1.2.1.1 Khái niệm:

Học liên tục (Continual Learning) là ý tưởng cập nhật mô hình của bạn khi dữ liệu mới trở nên có sẵn; điều này giúp mô hình của bạn điều chỉnh với các phân phối dữ liệu hiện tại.

Một khi mô hình của bạn đã được cập nhật, nó không thể được phát hành mù quáng vào sản xuất. Cần kiểm thử để đảm bảo rằng nó an toàn và tốt hơn mô hình hiện tại trong sản xuất.

1.2.1.2 Continual Learning thường bị hiểu lầm:

Học liên tục không chỉ đề cập đến một loại đặc biệt của các thuật toán học máy cho phép cập nhật từng điểm dữ liệu mới một khi nó trở nên có sẵn. Ví dụ về lớp thuật toán đặc biệt này là cập nhật Bayes tuần tự và bộ phân loại KNN. Lớp thuật toán nhỏ này đôi khi được gọi là "các thuật toán học trực tuyến".

Khái niệm của Học liên tục có thể được áp dụng vào bất kỳ thuật toán học máy giám sát nào, không chỉ là một lớp đặc biệt.

Học liên tục KHÔNG có nghĩa là bắt đầu một công việc đào tạo lại mỗi khi có một điểm dữ liệu mới xuất hiện. Trên thực tế, điều này nguy hiểm vì nó khiến cho các mạng thần kinh dễ bị quên nặng.

Hầu hết các công ty sử dụng học liên tục cập nhật mô hình của họ theo dạng "micro-batch" (ví dụ: mỗi 512 hoặc 1024 ví dụ). Số lượng ví dụ tối ưu phụ thuộc vào công việc cụ thể.

Học liên tục có vẻ như là một công việc của các nhà khoa học dữ liệu ở bề mặt. Tuy nhiên, thường xuyên, nó đòi hỏi nhiều công việc cơ sở hạ tầng để triển khai và duy trì hiệu suất cao. Thực tế, việc học liên tục tốn kém về mặt tính toán so với việc đào tạo lại mô hình một cách định kỳ từ đầu.

Trong nhiều trường hợp, học liên tục không có ý nghĩa. Bạn có thể sử dụng một mô hình đã được đào tạo từ trước và không cần phải cập nhật nó. Tùy thuộc vào tình huống, đôi khi làm mới mô hình có thể làm tăng độ phức tạp mà không cung cấp lợi ích đáng kể.

1.2.1.3 Thực hiện Continual Learning:

- Làm mới đặc trưng

Nếu bạn làm việc với các đặc trưng liên tục, bạn có thể muốn cập nhật các giá trị thống kê (ví dụ: trung bình, độ lệch chuẩn) khi có dữ liệu mới. Điều này có thể giúp mô hình của bạn đáp ứng nhanh chóng với các thay đổi trong dữ liệu.

- Làm mới mô hình
  - Nếu bạn đang sử dụng mô hình học máy giám sát (ví dụ: hồi quy hoặc phân loại), bạn có thể cần đào tạo lại mô hình của mình khi có dữ liệu mới. Quy trình này đòi hỏi quyết định cẩn thận vì nó có thể làm tăng độ phức tạp và chi phí tính toán.
  - Một cách để giảm chi phí là sử dụng phương pháp tăng tốc đào tạo, như tăng tốc đào tạo học máy (Học máy trực tuyến) hoặc các biến thể của SGD tối ưu hóa từng mini-batch.

1.2.2 Test Production

Kiểm thử mô hình (Test Production) trong môi trường sản xuất là một bước quan trọng để đảm bảo rằng mô hình được triển khai hoạt động hiệu quả và an toàn. Dưới đây là một số phương pháp phổ biến.

- A/B Testing (Thử nghiệm A/B)
  - Thử nghiệm A/B là một phương pháp so sánh hiệu suất giữa hai phiên bản của mô hình: phiên bản hiện tại (phiên bản A) và phiên bản mới (phiên bản B).
  - Dữ liệu được chia thành hai nhóm ngẫu nhiên, mỗi nhóm nhận một phiên bản khác nhau của mô hình.

  - Kết quả được so sánh để xác định xem phiên bản nào mang lại hiệu suất tốt hơn.
  - Phương pháp này yêu cầu lưu ý đến những ảnh hưởng không mong muốn và đảm bảo rằng sự chênh lệch giữa hai phiên bản không phải là ngẫu nhiên.

- Kiểm thử đa biến
  - Kiểm thử đa biến liên quan đến việc kiểm tra mô hình trong nhiều điều kiện khác nhau để đảm bảo nó hoạt động chính xác và hiệu quả trong mọi tình huống.
  - Các điều kiện khác nhau có thể bao gồm sự thay đổi trong dữ liệu đầu vào, môi trường hoạt động, hoặc thậm chí là kiến trúc hạ tầng.
- Kiểm thử tự động
  - Kiểm thử tự động liên quan đến việc tự động hóa quy trình kiểm thử mô hình.
  - Điều này có thể bao gồm việc triển khai các bộ kiểm thử tự động, việc theo dõi hiệu suất mô hình tự động, và tự động hóa việc triển khai mô hình mới.
- Giả mạo (Mocking)

Giả mạo là một kỹ thuật trong đó một số thành phần của môi trường sản xuất được thay thế bằng các thành phần giả mạo, giúp kiểm thử mô hình trong môi trường có kiểm soát hơn.

- Đối chiếu dữ liệu
  - Đối chiếu dữ liệu là quá trình so sánh dữ liệu mô hình thực tế với dữ liệu mà mô hình được đào tạo.
  - Điều này giúp phát hiện ra các sự chênh lệch trong phân phối dữ liệu và kiểm tra xem mô hình có đáp ứng tốt với sự thay đổi đó hay không.

1.2.3 Tổng kết

Học liên tục (Continual learning) và kiểm thử mô hình (Test production) trong môi trường sản xuất là hai khía cạnh quan trọng của việc duy trì và cập nhật mô hình trong môi trường thực tế. Kết hợp kỹ thuật học máy, quy trình kỹ thuật và công cụ hỗ trợ là chìa khóa để triển khai thành công các quy trình này. Cẩn thận và đánh giá kỹ lưỡng là quan trọng khi triển khai những thay đổi này để đảm bảo hiệu suất và an toàn của mô hình.

BÀI 2 – LÀM CHUNG TRONG NHÓM

Bảng phân công công việc:

https://docs.google.com/spreadsheets/d/1sP36em-pYKGAaeYSW2pBDH28KmmA3c9NoqonIdMmnFA/edit?usp=sharing

Tải dữ liệu đã có sẵn trên Kaggle:

https://www.kaggle.com/datasets/alopez247/pokemon/

Làm bài trên Google Colab:

https://colab.research.google.com/drive/1srakNfDMSPI5cDnjHQsjFjFYFcbgRxf7?usp=sharing

Link Github:

https://github.com/quynhnhu1692/Final-Project

TÀI LIỆU THAM KHẢO

Tiếng Việt

Tiếng Anh

[1] amananandrai, *10 famous Machine Learning Optimizers*, Machine Learning, https://dev.to/amananandrai/10-famous-machine-learning-optimizers-1e22

[2] serodriguez68, *Chapter 9 - Continual learning and test in production*, Machine Learning, https://github.com/serodriguez68/designing-ml-systems-summary/blob/main/09-continual-learning-and-test-in-production.md#chapter-9---continual-learning-and-test-in-production