

**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**

*Người hướng dẫn:* **PGS.TS Lê Anh Cường**

*Người thực hiện:* **Lương Quang Vinh - 52100865**

**Khoá : 25**

**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**

*Người hướng dẫn:* **PGS.TS Lê Anh Cường**

*Người thực hiện:* **Lương Quang Vinh - 52100865**

**Khoá : 25**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

## LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành đến thầy về những kiến thức quý báu và sự hướng dẫn tận tâm trong môn nhập môn học máy. Thầy đã giúp em hiểu rõ hơn về chủ đề này và trang bị cho em những kỹ năng quan trọng. Em hết sức biết ơn công lao của thầy và cam kết nỗ lực hết mình để hoàn thành tốt môn học. Mong rằng thầy sẽ tiếp tục hỗ trợ và khuyến khích em trong quá trình học tập.

Một lần nữa, em xin bày tỏ lòng biết ơn sâu sắc đến thầy.

Trân trọng,

## **ĐỒ ÁN ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là sản phẩm đồ án của riêng chúng tôi và được sự hướng dẫn của Thầy 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 6 tháng 12 năm 2023*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Lương Quang Vinh*

## 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

Môn Học máy cung cấp cho sinh viên các kiến thức nền tảng về máy học và ứng dụng của nó trong thực tế. Trong khóa học, chúng em được làm quen với các khái niệm cơ bản như training data, models, learning algorithms. Giảng viên cũng giới thiệu cho sinh viên một số thuật toán học máy phổ biến hiện nay như linear regression, logistic regression, decision trees, neural networks, v.v...

Ngoài ra, chúng em cũng được thực hành xây dựng mô hình học máy để giải quyết các bài toán thực tế như dự đoán, phân loại, tóm tắt văn bản, v.v... Thông qua đó, sinh viên có cơ hội ứng dụng lý thuyết để giải quyết các vấn đề trong thực tế với dữ liệu thực. Đồng thời rèn luyện kỹ năng lập trình với Python để xây dựng các mô hình Machine Learning.

## MỤC LỤC

LỜI CẢM ƠN .....	i
PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN .....	iii
TÓM TẮT .....	iv
MỤC LỤC.....	1
PHẦN 1 :.....	2
TÀI LIỆU THAM KHẢO.....	12

## PHẦN 1 :

Trình bày một bài nghiên cứu, đánh giá của em về các vấn đề sau:

- 1) Tìm hiểu, so sánh các phương pháp Optimizer trong huấn luyện mô hình học máy;
- 2) Tìm hiểu về Continual Learning và Test Production khi xây dựng một giải pháp học máy để giải quyết một bài toán nào đó.

### Câu 1:

- Optimizer ( thuật toán tối ưu ) là cơ sở để xây dựng mô hình neural network với mục đích học được các features của dữ liệu đầu vào, từ đó tìm được một cặp weights và bias phù hợp để tối ưu hóa model.
- Các phương pháp Optimizer:
  - Gradient Descent (GD) : "Gradient Descent" là một phương pháp tối ưu hóa được sử dụng để điều chỉnh các tham số của một mô hình máy học để giảm thiểu hàm mất mát (loss function).
    - Ưu điểm: thuật toán cơ bản, dễ hiểu. Thuật toán đã giải quyết được vấn đề tối ưu model neural network bằng cách cập nhật trọng số sau mỗi vòng lặp
    - Nhược điểm: vì thuật toán còn đơn giản nên còn nhiều hạn chế, như phụ thuộc vào nghiệm khởi tạo ban đầu và learning rate.
  - Stochastic Gradient Descent (SGD) : stochastic là một biến thể của Gradient Descent. SGD là một thuật toán tối ưu hóa được sử dụng rộng rãi trong máy học và học sâu. Thay vì tính toán Gradient dựa trên toàn bộ tập dữ liệu (batch gradient descent), SGD chỉ tính toán gradient trên một phần ngẫu nhiên của dữ liệu. Sau đó cập nhật tham số mô hình theo hướng giảm dần của hàm loss
    - Ưu điểm :



- Tốc độ huấn luyện nhanh do chỉ cần tính toán trên một mẫu nhỏ
- Khả năng thoát khỏi các điểm cực trị cục bộ cực tốt
- Kiểm soát độ lớn của thay đổi hàm số tốt hơn
- Nhược điểm :
  - Dễ bị dao động do tính ngẫu nhiên của gradient
  - Khó hội tụ vào điểm cực tiểu toàn cục
- Adagrad : Không giống như các thuật toán trước đó thì learning rate hầu như giống nhau trong quá trình training (learning rate là hằng số), Adagrad coi learning rate là 1 tham số. Tức là Adagrad sẽ cho learning rate biến thiên sau mỗi thời điểm t.

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

- Trong đó :
  - $\eta$  : hằng số
  - $g_t$  : gradient tại thời điểm t
  - $\epsilon$  : 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 vector tham số tại thời điểm t.
- Ưu điểm : Một lợi ích dễ thấy của Adagrad là tránh việc điều chỉnh learning rate bằng tay, chỉ cần để tốc độ học default là 0.01 thì thuật toán sẽ tự động điều chỉnh.
- Nhược điểm : Yếu điểm của Adagrad là 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.

- RMSprop : RMSprop giải quyết vấn đề tỷ lệ học giảm dần của Adagrad bằng cách chia tỷ lệ học cho trung bình của bình phương gradient.

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

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

- Trong đó:
  - $\theta_t$  là vector trọng số tại bước thời gian  $t$ .
  - $g_t$  là gradient của hàm mất mát tại bước thời gian  $t$ .
  - $E[g^2]_t$  là trung bình độ vuông của gradient, được tính với trọng số giảm dần theo thời gian.
  - $\eta$  là learning rate.
  - $\epsilon$  là một số nhỏ (ví dụ:  $1 \times 10^{-8}$  đến  $1 \times 10^{-8}$ ) được thêm vào trong mẫu số để tránh chia cho 0.

- Ưu điểm : Ưu điểm rõ nhất của RMSprop là giải quyết được vấn đề tốc độ học giảm dần của Adagrad ( vấn đề tốc độ học giảm dần theo thời gian sẽ khiến việc training chậm dần, có thể dẫn tới bị đóng băng )
- Nhược điểm : Vẫn có thể gặp vấn đề với việc giảm learning rate quá nhanh.

- Momentum : là một phương pháp tối ưu hóa được sử dụng trong quá trình đào tạo mô hình trong Machine Learning và Deep Learning. Mục tiêu của thuật toán này là giảm bớt tình trạng dao động đặc biệt khi mô hình đang tiến hành cập nhật trọng số. Nó được thiết kế để giữ cho mô

hình di chuyển nhanh chóng về hướng đúng và tránh được mắc kẹt ở các điểm tối thiểu cục bộ.

- $x_{\text{new}} = x_{\text{old}} - (\text{gama} \cdot v + \text{learningrate} \cdot \text{gradient})$ 
  - Trong đó :  $x_{\text{new}}$ : tọa độ mới
    - $x_{\text{old}}$  : tọa độ cũ
    - $\text{gama}$ : parameter , thường =0.9
    - $\text{learningrate}$  : tốc độ học
    - $\text{gradient}$  : đạo hàm của hàm  $f$
- Ưu điểm: tăng tốc độ học
  - Giảm dao động
  - Vượt qua thung lũng
  - Ít bị mắc kẹt
- Nhược điểm : Overshooting
- Adam : Adam là sự kết hợp của Momentum và RMSprop . Nếu giải thích theo hiện tượng vật lí thì Momentum giống như 1 quả cầu lao xuống dốc, còn Adam như 1 quả cầu rất nặng có ma sát, vì vậy nó dễ dàng vượt qua local minimum tới global minimum và khi tới global minimum nó không mất nhiều thời gian dao động qua lại quanh đích vì nó có ma sát nên dễ dừng lại hơn.
  - Công thức cập nhật cho trung bình của gradient (momentum):
 
$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$$

- Công thức cập nhật cho trung bình của bình phương gradient (độ đồng nhất):  $v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$
- Điều chỉnh trung bình gradient và trung bình bình phương gradient:  $m^t = \frac{mt}{1 - b_1}$ 
  - $v^t = \frac{vt}{1 - b_2}$
- Công thức cập nhật tham số:  $\theta_t = \theta_{t-1} - v^t + \epsilon \alpha \cdot m^t$ 
  - Trong đó:
    - $mt$  là trung bình của gradient tại thời điểm  $t$ .
    - $vt$  là trung bình của bình phương gradient tại thời điểm  $t$ .
    - $gt$  là gradient tại thời điểm  $t$ .
    - $\beta_1$  và  $\beta_2$  là các hệ số giảm môment và giảm độ đồng nhất, thường được đặt là 0.9 và 0.999.
    - $\alpha$  là tỷ lệ học (learning rate).
    - $\epsilon$  là một hằng số nhỏ (thường là  $10^{-8}$ ) để tránh chia cho 0.
- Ưu điểm : hiệu suất cao trên nhiều loại bài toán
  - Tự động điều chỉnh tỷ lệ học
  - Ít cần thiết lập siêu tham số

- Thích hợp cho các bài toán với dữ liệu lớn và nhiều
  - Nhược điểm : có thể dẫn đến overfitting
  - Khó khăn trong việc điều chỉnh siêu tham số
  - Yêu cầu lưu trữ thêm bộ nhớ
- So sánh ưu nhược điểm của các phương pháp trên :

Phương pháp	Ưu điểm	Nhược điểm	Bài toán phù hợp
Gradient Descent (GD)	<ul style="list-style-type: none"> <li>- Đơn giản và dễ hiểu.</li> <li>- Không cần lưu trữ nhiều thông tin, phù hợp với dữ liệu lớn.</li> </ul>	<ul style="list-style-type: none"> <li>- Chậm trên dữ liệu lớn vì cần tính toán gradient trên toàn bộ tập dữ liệu.</li> <li>- Dễ bị rơi vào điểm tối thiểu cục bộ.</li> </ul>	Phù hợp với bài toán có tập dữ liệu nhỏ và không quá phức tạp.
SGD (Stochastic Gradient Descent)	<ul style="list-style-type: none"> <li>- Hiệu quả tính toán hơn đối với dữ liệu lớn.</li> <li>- Có thể thoát khỏi các điểm cực tiểu địa phương.</li> </ul>	<ul style="list-style-type: none"> <li>- Không ổn định, dao động nhiều.</li> <li>- Cần điều chỉnh kích thước batch và tốc độ học.</li> </ul>	Thích hợp với dữ liệu lớn và bài toán phức tạp.
Momentum	<ul style="list-style-type: none"> <li>- Giảm dao động và giúp vượt qua điểm cực tiểu địa phương.</li> <li>- Hội tụ nhanh hơn so với GD.</li> </ul>	<ul style="list-style-type: none"> <li>- Có thể quá chậm hoặc quá nhanh tùy thuộc vào siêu tham số.</li> </ul>	Phù hợp với bài toán có địa hình phức tạp.
RMSprop	<ul style="list-style-type: none"> <li>- Hiệu quả cho các mô hình không đồng nhất.</li> <li>- Tự động điều chỉnh tốc độ học.</li> </ul>	<ul style="list-style-type: none"> <li>- Cần lựa chọn siêu tham số.</li> </ul>	Phù hợp với bài toán có dữ liệu không đồng nhất.
Adagrad	<ul style="list-style-type: none"> <li>- Tích hợp tốt với các bài toán có các đặc trưng hiếm gặp.</li> <li>- Không cần thiết lập tốc độ học.</li> </ul>	<ul style="list-style-type: none"> <li>- Dễ phát sinh vấn đề về độ giảm (learning rate decay).</li> <li>- Không phù hợp cho các bài toán với</li> </ul>	Phù hợp với bài toán có dữ liệu không đồng đều.

		đặc trưng thường xuyên xuất hiện.	
Adam	<ul style="list-style-type: none"> <li>- Kết hợp ưu điểm của Momentum và RMSprop.</li> <li>- Hiệu quả với dữ liệu lớn và có tính hiệu quả cao</li> </ul>	- Cần lựa chọn siêu tham số.	Phù hợp với nhiều loại bài toán, đặc biệt là trong học sâu.

**Câu 2 :** 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**

Continual Learning (hoặc Lifelong Learning) là một lĩnh vực trong Machine Learning (ML) mà mô hình được huấn luyện liên tục trên dữ liệu mới mà nó chưa từng thấy trước đó. Điều này đặc biệt quan trọng trong các ứng dụng thực tế nơi dữ liệu thường xuyên thay đổi theo thời gian. Mục tiêu của Continual Learning là giữ cho mô hình cập nhật với kiến thức mới mà không quên đi kiến thức cũ.

Một số thách thức chính trong Continual Learning bao gồm:

- Lãng quên (Catastrophic Forgetting): Mô hình có thể quên thông tin về tác vụ trước đó khi học tác vụ mới.
- Interference: Học một tác vụ mới có thể làm ảnh hưởng đến khả năng của mô hình đối với các tác vụ đã học trước đó.
- Rò rỉ thông tin (Information Leakage): Kiến thức từ tác vụ mới có thể rò rỉ vào mô hình và làm ảnh hưởng đến khả năng dự đoán của nó trên các tác vụ trước đó.
- Các phương pháp trong Continual Learning bao gồm sử dụng bộ nhớ ngoại vi, regularization techniques, và mô hình hóa nguyên tắc học.

## Test Production

Test Production là quá trình tạo ra các tập kiểm thử (test sets) cho mô hình học máy. Trong quá trình phát triển mô hình, chúng ta thường chia dữ liệu thành tập huấn luyện và tập kiểm thử để đánh giá hiệu suất của mô hình trên dữ liệu mới mà nó chưa thấy. Quá trình tạo tập kiểm thử là quan trọng để đảm bảo rằng mô hình không chỉ học thuộc lòng dữ liệu huấn luyện mà còn có khả năng tổng quát hóa cho dữ liệu mới.

Một số điểm cần xem xét khi tạo tập kiểm thử:

- Đại diện (Representativeness): Tập kiểm thử cần phản ánh phân phối của dữ liệu thực tế để mô hình được đánh giá một cách đúng đắn.
- Khả năng phân loại (Coverage): Tập kiểm thử cần bao phủ tất cả các trường hợp quan trọng và biểu đồ đa dạng của dữ liệu.
- Khả năng khám phá (Exploratory Power): Tập kiểm thử cần chứa các điều kiện đặc biệt và tình huống mà mô hình có thể gặp trong ứng dụng thực tế.
- Thay đổi định kỳ (Periodic Update): Cập nhật tập kiểm thử theo thời gian để đảm bảo rằng nó vẫn đại diện cho dữ liệu mới và thách thức mới.

Khi tạo tập kiểm thử, quan trọng để duy trì sự cân bằng giữa tỷ lệ các lớp (nếu là một bài toán phân loại), tránh lạc quan về hiệu suất mô hình dựa trên một tập kiểm thử không đại diện.





## TÀI LIỆU THAM KHẢO

[1]. Các phương pháp Optimizer

<https://viblo.asia/p/optimizer-hieu-sau-ve-cac-thuat-toan-toi-uu-gdsgdadam-Qbq5QQ9E5D8>

[2]. Tham khảo cách sử dụng JQuery để trình bày, lọc danh sách sản phẩm

<https://www.youtube.com/watch?v=sHu8Est6cnc&t=2s>

[3]. Làm giỏ hàng bằng Session

<https://www.youtube.com/watch?v=JjoEmq84RGQ>