TRƯỜNG ĐẠI HỌC ĐIỆN LỰC

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO CHUYÊN ĐỀ HỌC PHẦN**

**HỌC SÂU**

**ĐỀ TÀI:**

**SỬ DỤNG MÔ HÌNH MẠNG NƠ RON HỒI QUY (RNN)**

**CHO BÀI TOÁN PHÂN LOẠI VĂN BẢN**

|  |  |
| --- | --- |
|  |  |
| |  |  | | --- | --- | | **Sinh viên thực hiện** | **: Ngô Minh Hiếu** | | **Giảng viên hướng dẫn** | **: TS.Phạm Đức Hồng** | | | **Mã sinh viên** | **: 21810310031** | | | **Lớp** | **: D16TTNT&TGMT** | | |  |  | | |  |
|  |  |

***Hà Nội, Tháng 4 năm 2024***

**PHIẾU CHẤM ĐIỂM**

**Sinh viên thực hiện:**

|  |  |  |
| --- | --- | --- |
| **Họ và tên** | **Chữ kí** | **Ghi chú** |
| **Ngô Minh Hiếu** |  |  |
|  |  |  |

**Giảng Viên Chấm:**

|  |  |  |
| --- | --- | --- |
| **Họ và tên** | **Chữ kí** | **Ghi chú** |
| **1.** |  |  |
| **2.** |  |  |

**MỤC LỤC**

[LỜI NÓI ĐẦU 1](#_Toc165577135)

[CHƯƠNG 1: TỔNG QUAN VỀ SUPPORT VECTOR MACHINE 2](#_Toc165577136)

[1.1. Phát biểu bài toán 2](#_Toc165577137)

[1.1.1. Trình bày tóm tắt phân lớp dữ liệu 3](#_Toc165577138)

[1.1.2. Tại sao sử dụng thuật toán SVM trong phân lớp dữ liệu 4](#_Toc165577139)

[1.2. THUẬT TOÁN SVM 5](#_Toc165577140)

[1.2.1. Giới thiệu 5](#_Toc165577141)

[1.2.2. Định nghĩa 5](#_Toc165577142)

[1.2.3. Ý tưởng của phương pháp 5](#_Toc165577143)

[1.2.4. Nội dung của phương pháp 6](#_Toc165577144)

[1.2.4.1. Cơ sở lý thuyết 6](#_Toc165577145)

[1.2.4.2. Bài toán phân 2 lớp SVM 7](#_Toc165577146)

[1.2.4.3. Bài toán nhiều phân lớp với SVM 7](#_Toc165577147)

[1.2.5. Các bước chính của phương pháp SVM 8](#_Toc165577148)

[1.2.6. Ưu điểm và nhược điểm của SVM 8](#_Toc165577149)

[1.2.7. Ứng dụng của SVM 9](#_Toc165577150)

[CHƯƠNG 2 ỨNG DỤNG THỰC TẾ 10](#_Toc165577151)

[2.1. Giới thiệu bài toán 10](#_Toc165577152)

[2.2. Giải quyết bài toán 11](#_Toc165577153)

[2.2.1. Nhập, xuất, phân tích dữ liệu 11](#_Toc165577154)

[2.2.2. Phân chia dữ liệu huấn luyện và kiểm thử 17](#_Toc165577155)

[2.2.3. Xây dựng mô hình, huấn luyện mô hình 18](#_Toc165577156)

[2.2.4. Đánh giá kết quả dự đoán 19](#_Toc165577157)

[KẾT LUẬN 25](#_Toc165577158)

# LỜI NÓI ĐẦU

Trong thập kỷ qua, lĩnh vực Học sâu (Deep Learning) đã nổi lên như một làn sóng mạnh mẽ trong cộng đồng nghiên cứu khoa học máy tính và trí tuệ nhân tạo. Khả năng của các mô hình học sâu trong việc tự động học và hiểu được dữ liệu phức tạp đã mang lại những tiến bộ đáng kinh ngạc, từ nhận dạng hình ảnh và giọng nói đến xử lý ngôn ngữ tự nhiên và tự động lái xe. Với sự phát triển nhanh chóng và tiềm năng không giới hạn, học sâu không chỉ là một lĩnh vực nghiên cứu quan trọng mà còn là một công cụ mạnh mẽ cho các ứng dụng thực tiễn trong nhiều lĩnh vực, từ công nghệ thông tin đến y học và tự động hóa công nghiệp.

Trong lĩnh vực học sâu, mạng nơ-ron hồi quy (RNN) đã trở thành một công cụ mạnh mẽ trong việc xử lý dữ liệu tuần tự như văn bản, âm thanh và chuỗi thời gian. Trên cơ sở khả năng của nó trong việc xử lý dữ liệu có cấu trúc thời gian, RNN đã được áp dụng rộng rãi trong nhiều ứng dụng, từ dịch thuật máy tính đến nhận dạng giọng nói và phân loại văn bản. Việc áp dụng mạng nơ-ron hồi quy trong phân loại văn bản hứa hẹn mang lại những kết quả tích cực và ứng dụng rộng rãi trong thực tế.

Để có thể hoàn thành báo cáo, em xin phép gửi lời cảm ơn chân thành tới thầy Phạm Đức Hồng đã truyền đạt, giảng dạy cho em những kiến thức, những kinh nghiệm quý báu trong thời gian học tập, tận tình hướng dẫn chúng em trong quá trình hoàn thành báo cáo này.

Em xin chân thành cảm ơn!

**CHƯƠNG 1. TỔNG QUAN VÈ HỌC SÂU**

**1.1. Khái niệm về học sâu**

**1.1.1. Học sâu là gì?**

Deep Learning hay học sâu là một tập hợp con của học máy, về cơ bản là một mạng nơ-ron có ba lớp trở lên. Những mạng lưới thần kinh này cố gắng mô phỏng hành vi của não người cho phép deep learning “học” từ một lượng lớn dữ liệu. Trong khi mạng nơ-ron có một lớp vẫn có thể đưa ra các dự đoán gần đúng, các lớp ẩn bổ sung có thể giúp tối ưu hóa và tinh chỉnh để có độ chính xác.

**1.1.2. Cách thức hoạt động của mô hình học sâu**

Mạng thần kinh nhân tạo trong Deep Learning được xây dựng để mô phỏng khả năng tư duy của bộ não con người. Cách thức hoạt động như sau:

* Một mạng nơ-ron bao gồm nhiều lớp (layer) khác nhau, càng nhiều lớp thì mạng sẽ càng “sâu”. Trong mỗi lớp là các node (nút mạng) và được liên kết với các lớp khác bên cạnh.
* Mỗi kết nối giữa các nút sẽ có một trọng số tương ứng, trọng số càng cao thì mức độ ảnh hưởng của kết nối này đến mạng nơ-ron càng lớn.
* Mỗi nơ-ron sẽ có một chức năng kích hoạt, về cơ bản chịu trách nhiệm "chuẩn hóa" đầu ra từ nơ-ron này.
* Dữ liệu được người dùng nhập vào mạng thần kinh đi qua tất cả các lớp và trả về kết quả ở layer cuối cùng, được gọi là lớp đầu ra (output layer).
* Trong quá trình đào tạo mô hình mạng nơ-ron, các trọng số sẽ được thay đổi và nhiệm vụ của mô hình là tìm tập giá trị của các trọng số sao cho phán đoán đúng nhất.

Các hệ thống Deep Learning yêu cầu phần cứng rất mạnh để có thể xử lý lượng lớn dữ liệu và thực hiện các phép tính phức tạp. Nhiều mô hình Deep Learning có thể mất hàng tuần hoặc thậm chí hàng tháng để triển khai trên phần cứng tiên tiến nhất hiện nay.

**1.1.3. Ưu nhược điểm của mô hình học sâu**

Deep Learning là một bước ngoặt lớn trong lĩnh vực trí tuệ nhân tạo - artificial intelligence. Nó cho phép các nhà khoa học dữ liệu khác xây dựng nhiều mô hình có độ chính xác cao trong các lĩnh vực nhận dạng hình ảnh, xử lý ngôn ngữ tự nhiên, data…

Một số ưu điểm nổi bật của Deep Learning bao gồm:

* Cấu trúc neural networks linh hoạt, dễ dàng thay đổi để phù hợp với nhiều thuật toán khác nhau.
* Có khả năng giải nhiều vấn đề phức tạp với độ chính xác rất cao.
* Khả năng tự động hóa cao, tự điều chỉnh và tự tối ưu hóa.
* Có khả năng thực hiện tính toán song song, hiệu năng tốt, xử lý lượng dữ liệu lớn.

Bên cạnh ưu điểm, Deep Learning vẫn còn tồn tại nhiều hạn chế:

* Cần khối lượng dữ liệu khổng lồ để khai thác tối đa khả năng của Deep Learning.
* Chi phí tính toán cao vì phải xử lý nhiều mô hình phức tạp.
* Không có nền tảng lý thuyết vững chắc để chọn các công cụ tối ưu cho Deep Learning.

**1.2. Ứng dụng thực tiễn của mô hình học sâu**

Deep Learning được ứng dụng trong các công việc đòi hỏi khả năng tính toán cao, xử lý dữ liệu lớn và độ phức tạp lớn. Dưới đây là một số ứng dụng phổ biến nhất của Deep Learning trên thực tế:

* Hệ thống xe tự lái

Một trong những công nghệ mới và thú vị nhất hiện nay là hệ thống lái xe tự động, được xây dựng trên các mạng thần kinh cấp cao. Nói một cách đơn giản, các mô hình Deep Learning sẽ nhận diện các vật thể trong môi trường xung quanh xe.

Tiếp đến mô hình sẽ tiến hành tính toán khoảng cách giữa xe và các phương tiện khác, xác định tín hiệu đèn giao thông, làn đường quy định… Từ đó đưa ra các quyết định tối ưu nhất và nhanh nhất.

* Phân tích cảm xúc

Đây là lĩnh vực phân tích cảm xúc của con người thông qua xử lý ngôn ngữ tự nhiên, phân tích văn bản và thống kê. Doanh nghiệp có thể ứng dụng Deep Learning để hiểu và phán đoán cảm xúc của khách hàng dựa trên các đánh giá, bình luận, feedback… từ đó đưa ra các chiến lược kinh doanh, marketing phù hợp với từng phân khúc khách hàng.

* Mạng xã hội

Một số nền tảng mạng xã hội lớn như Twitter cũng áp dụng thuật toán Deep Learning để cải thiện dịch vụ của họ. Cụ thể, các trang web này sẽ phân tích lượng lớn dữ liệu thông qua mạng thần kinh nhân tạo để tìm hiểu về sở thích và xu hướng hiện tại của người dùng.

Ngoài ra, Instagram hay facebook cũng sử dụng Deep Learning để tránh các hành vi bạo lực trên không gian mạng. Chặn các bình luận xúc phạm, vi phạm tiêu chuẩn cộng đồng…

* Trợ lý ảo - virtual assistant

Trợ lý ảo đang được ứng dụng rất nhiều trong cuộc sống hàng ngày. Trong đó phổ biến phải kể đến chatbot, Google Assistant, Cortana. Siri, ... Các trợ lý này được xây dựng dựa trên mô hình Deep Learning với các thuật toán giúp nhận dạng và xử lý dữ liệu như giọng nói, văn bản…

* Lĩnh vực chăm sóc sức khỏe

Deep Learning cũng có đóng góp đáng kể cho lĩnh vực y tế, trong đó các mô hình phổ biến bao gồm mô hình dự đoán bệnh, chẩn đoán ung thư, phân tích kết quả chụp MRI, X-quang…

**CHƯƠNG 2. GIỚI THIỆU VỀ MẠNG NƠ RON HỒI QUY (RNN)**

**2.1. Khái niệm về mạng nơ ron hồi quy (RNN)**

**2.1.1. Khái niệm**

Ý tưởng chính của RNN (Recurrent Neural Network) là sử dụng chuỗi các thông tin. Trong các mạng nơ-ron truyền thống tất cả các đầu vào và cả đầu ra là độc lập với nhau. Tức là chúng không liên kết thành chuỗi với nhau. Nhưng các mô hình này không phù hợp trong rất nhiều bài toán. RNN được gọi là hồi quy (Recurrent) bởi lẽ chúng thực hiện cùng một tác vụ cho tất cả các phần tử của một chuỗi với đầu ra phụ thuộc vào cả các phép tính trước đó. Nói cách khác, RNN có khả năng nhớ các thông tin được tính toán trước đó. Trên lý thuyết, RNN có thể sử dụng được thông tin của một văn bản rất dài, tuy nhiên thực tế thì nó chỉ có thể nhớ được một vài bước trước. Về cơ bản một mạng RNN có dạng như sau:



Hình 2.1. Hình dạng của mạng nơ ron hồi quy RNN

Mô hình trên mô tả phép triển khai nội dung của một RNN. Triển khai ở đây có thể hiểu đơn giản là ta vẽ ra một mạng nơ-ron chuỗi tuần tự. Ví dụ ta có một câu gồm 5 chữ “Đẹp trai lắm gái theo”, thì mạng nơ-ron được triển khai sẽ gồm 5 tầng nơ-ron tương ứng với mỗi chữ một tầng. Lúc đó việc tính toán bên trong RNN được thực hiện như sau:

là đầu vào tại bước . Ví dụ, là một vec-tơ one-hot tương ứng với từ thứ 2 của câu (trai).

là trạng thái ẩn của bước . Nó chính là bộ nhớ của mạng. được tính toán dựa trên cả các trạng thái ẩn phía trước và đầu vào tại bước đó: . Hàm thường là một hàm phi tuyến tính như tang hyperbolic (tanh) hay ReLu. Để làm phép toán cho phần tử ẩn đầu tiên ta cần khởi tạo thêm , thường giá trị khởi tạo được gắn bằng 0.

là đầu ra từ bước *t.* Ví dụ, ta muốn dự đoán từ tiếp theo có thể xuất hiện trong câu thì chính là một vec-tơ xác xuất các từ trong danh sách từ vựng của ta: .

**2.1.2. Cách hoạt động của mạng hồi quy (RNN)**



*Hình 2.2. Sơ đồ của một mạng hồi quy RNN*

RNN được tạo thành từ các nơ-ron: các nút xử lý dữ liệu kết hợp cùng nhau để thực hiện các tác vụ phức tạp. Các nơ-ron được tổ chức dưới dạng lớp đầu vào, đầu ra và ẩn. Lớp đầu vào nhận thông tin để xử lý và lớp đầu ra cung cấp kết quả. Quá trình xử lý dữ liệu, phân tích và dự đoán diễn ra trong lớp ẩn.

* Lớp ẩn

RNN hoạt động bằng cách lần lượt truyền dữ liệu tuần tự nhận được đến các lớp ẩn. Tuy nhiên, RNN cũng có quy trình làm việc tự lặp lại hay hồi quy: lớp ẩn có thể ghi nhớ và sử dụng các đầu vào trước đó cho các dự đoán trong tương lai trong một thành phần bộ nhớ ngắn hạn. Quy trình này sử dụng đầu vào hiện tại và bộ nhớ đã lưu trữ để dự đoán chuỗi tiếp theo.

* Đào tạo

Các kỹ sư máy học (ML) đào tạo các mạng nơ-ron sâu như RNN bằng cách cung cấp dữ liệu đào tạo cho mô hình và tinh chỉnh hiệu năng của mô hình. Trong ML, trọng số của nơ-ron là tín hiệu để xác định mức độ ảnh hưởng của thông tin đã học trong quá trình đào tạo khi dự đoán đầu ra. Mỗi lớp trong RNN đều có trọng số bằng nhau.

Các kỹ sư ML điều chỉnh trọng số để dự đoán chính xác hơn. Họ sử dụng một kỹ thuật gọi là truyền ngược qua thời gian (BPTT) để tính lỗi mô hình và điều chỉnh trọng số của mô hình cho phù hợp. BPTT khôi phục đầu ra về bước thời gian trước và tính lại tỷ lệ lỗi. Qua đó, kỹ thuật này có thể xác định trạng thái ẩn nào trong chuỗi đang gây ra lỗi đáng kể và điều chỉnh lại trọng số để giảm biên lỗi.

**2.1.3. Phân loại các mạng nơ ron hồi quy (RNN)**

RNN thường có đặc trưng là kiến trúc một-một: một chuỗi đầu vào được liên kết với một đầu ra. Tuy nhiên, có thể điều chỉnh linh hoạt thành các cấu hình khác nhau cho các mục đích cụ thể. Sau đây là một số loại RNN phổ biến:

* Một-nhiều: Loại RNN này dẫn một đầu vào đến một số đầu ra. Loại này tạo điều kiện cho các ứng dụng ngôn ngữ như chú thích hình ảnh bằng cách tạo một câu từ một từ khóa duy nhất.
* Nhiều-nhiều: Mô hình sử dụng nhiều đầu vào để dự đoán nhiều đầu ra. Ví dụ: bạn có thể tạo một công cụ dịch ngôn ngữ bằng RNN, với khả năng phân tích câu và cấu trúc chính xác các từ trong một ngôn ngữ khác.
* Nhiều-một: Một số đầu vào được ánh xạ đến một đầu ra. Loại này rất hữu ích trong các ứng dụng như phân tích cảm xúc, trong đó mô hình dự đoán cảm xúc của khách hàng như tích cực, tiêu cực và trung lập từ lời chứng thực đầu vào.

**2.1.4. Những hạn chế của mạng nơ-ron hồi quy (RNN)**

Kể từ khi RNN được đưa vào sử dụng, các kỹ sư ML đã đạt được tiến bộ đáng kể trong các ứng dụng xử lý ngôn ngữ tự nhiên (NLP) bằng RNN cùng các biến thể. Tuy nhiên, bộ mô hình RNN có một số hạn chế:

* Độ dốc cực lớn

RNN có thể dự đoán sai đầu ra trong khóa đào tạo ban đầu. Bạn cần lặp lại nhiều lần để điều chỉnh các thông số của mô hình nhằm giảm tỷ lệ lỗi. Bạn có thể mô tả độ nhạy của tỷ lệ lỗi tương ứng với thông số của mô hình dưới dạng độ dốc. Bạn có thể hình dung độ dốc như một đường dốc mà bạn đi xuống từ một ngọn đồi. Độ dốc lớn hơn cho phép mô hình học nhanh hơn và độ dốc nhỏ làm giảm tốc độ học tập.

Độ dốc cực lớn xuất hiện khi độ dốc tăng theo cấp số nhân cho đến khi RNN trở nên không ổn định. Khi độ dốc trở nên lớn vô hạn, RNN hoạt động thất thường, dẫn đến các vấn đề về hiệu năng như quá khớp. Quá khớp là hiện tượng mô hình có thể dự đoán chính xác với dữ liệu đào tạo nhưng không thể dự đoán chính xác với dữ liệu thực tế.

* Độ dốc biến mất

Bài toán độ dốc biến mất là một điều kiện, trong đó độ dốc của mô hình đạt đến 0 trong quá trình đào tạo. Khi độ dốc biến mất, RNN không học hiệu quả từ dữ liệu đào tạo, dẫn đến chưa khớp. Mô hình chưa khớp không thể hoạt động tốt trong các ứng dụng thực tế vì các trọng số chưa được điều chỉnh thích hợp. RNN sẽ có rủi ro gặp phải vấn đề độ dốc cực lớn và biến mất khi xử lý các chuỗi dữ liệu dài.

* Thời gian đào tạo chậm

RNN xử lý dữ liệu tuần tự, do đó hạn chế khả năng xử lý khối lượng lớn văn bản một cách hiệu quả. Ví dụ: mô hình RNN có thể phân tích cảm xúc của người mua từ một vài câu. Tuy nhiên, mô hình này yêu cầu phải có năng lực điện toán cực lớn, không gian bộ nhớ và thời gian để tóm tắt một trang của một bài luận.

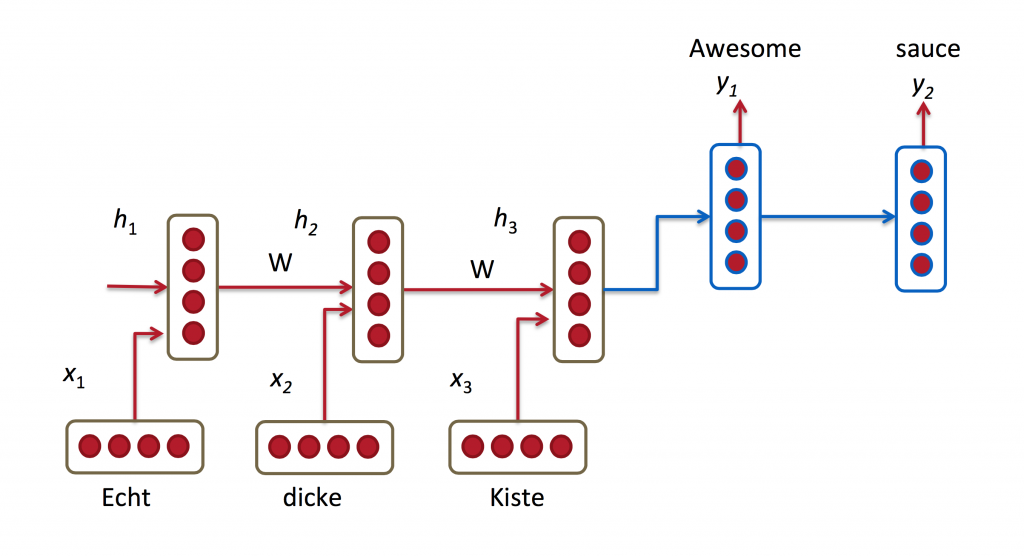
**2.2. Khả năng của RNN**

* Mô hình hóa ngôn ngữ và sinh văn bản

Mô hình ngôn ngữ cho phép ta dự đoán được xác xuất của một từ nào đó xuất hiện sau một chuỗi các từ đi liền trước nó. Do có khả năng ước lượng được độ tương tự của các câu nên nó còn được ứng dụng cho việc dịch máy. Một điểm lý thú của việc có thể dự đoán được từ tiếp theo là ta có thể xây dựng được một mô hình tự sinh từ cho phép máy tính có thể tự tạo ra các văn bản mới từ tập mẫu và xác xuất đầu ra của mỗi từ. Vậy nên, tùy thuộc vào mô hình ngôn ngữ mà ta có thể tạo ra được nhiều văn bản khác nhau khá là thú vị phải không. Trong mô hình ngôn ngữ, đầu vào thường là một chuỗi các từ (được mô tả bằng vec-tơ one-hot) và đầu ra là một chuỗi các từ dự đoán được. Khi huấn luyện mạng, ta sẽ gán vì ta muốn đầu ra tại bước chính là từ tiếp theo của câu.

* Dịch máy

Dịch máy (Machine Translation) tương tự như mô hình hóa ngôn ngữ ở điểm là đầu vào là một chuỗi các từ trong ngôn ngữ nguồn (ngôn ngữ cần dịch - ví dụ là tiếng Việt). Còn đầu ra sẽ là một chuỗi các từ trong ngôn ngữ đích (ngôn ngữ dịch - ví dụ là tiếng Anh). Điểm khác nhau ở đây là đầu ra của ta chỉ xử lý sau khi đã xem xét toàn bộ chuỗi đầu vào. Vì từ dịch đầu tiên của câu dịch cần phải có đầy đủ thông tin từ đầu vào cần dịch mới có thể suy luận được.



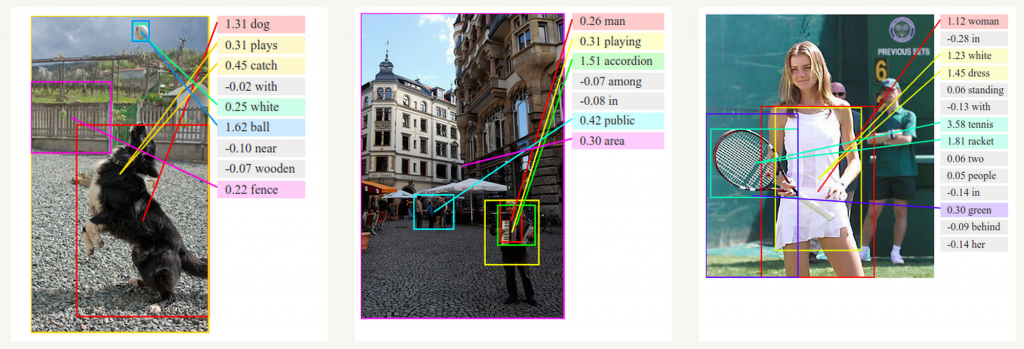
*Hình 2.3. Mạng hồi quy (RNN) cho ứng dụng dịch máy*

* Nhận dạng giọng nói

Đưa vào một chuỗi các tín hiệu âm thanh, ta có thể dự đoán được chuỗi các đoạn ngữ âm đi kèm với xác xuất của chúng.

* Mô tả hình ảnh

Cùng với ConvNet, RNN được sử dụng để tự động tạo mô tả cho các ảnh chưa được gán nhãn. Sự kết hợp này đã đưa ra được các kết quả khá kinh ngạc. Ví dụ như các ảnh dưới đây, các mô tả sinh ra có mức độ chính xác và độ tường tận khá cao.



*Hình 2.4. Sự sắp xếp ngữ nghĩa-hình ảnh sâu sắc để tạo mô tả hình ảnh*

**CHƯƠNG 3. ỨNG DỤNG MÔ HÌNH MẠNG NƠ RON HỒI QUY**

# 

# KẾT LUẬN

Bài toán phân lớp là một bài toán quan trọng trong support vector machine. Mục tiêu của bài toán là phân loại các điểm dữ liệu vào các lớp khác nhau dựa trên các đặc trưng của chúng. Bài toán phân lớp yêu cầu xác định các đặc trưng quan trọng để phân biệt giữa các lớp. Quá trình trích xuất đặc trưng từ dữ liệu gốc có thể là một bước quan trọng để cải thiện hiệu suất phân loại. Với phần triển khai, em sử dụng thư viện chính là scikit-learn.

Song, qua quá trình tìm hiểu và xây dựng mô hình, em nhận thấy chương trình của chúng em có những ưu nhược điểm như sau:

* Ưu điểm: chương trình được xây dựng một cách nhanh chóng, tuần tự, rõ ràng các bước, có sự trực quan để có thể dễ dàng hiểu về dữ liệu.
* Nhược điểm: chương trình còn đơn giản, độ chính xác của mô hình chưa cao, cần đi sâu và điều chỉnh mô hình cho phù hợp với bài toán hơn.

Với lượng kiến thức còn hạn hẹp, em rất mong nhận được nhận xét của các thầy cô cho bài báo cáo này để chúng em có thể đi sâu và hoàn thành tốt hơn về đề tài của mình.

Trong thời gian tới, em sẽ tiếp tục nâng cấp và hoàn thiện nhằm nâng cao tỉ lệ chính xác để giải quyết bài toán một cách nhanh gọn, tiết kiệm chi phí tối đa và dữ liệu được sử dụng một cách có ích.