**- ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KHOA HỌC VÀ KỸ THUẬT THÔNG TIN**



**BÁO CÁO BÀI TẬP TÌM HIỂU THUẬT TOÁN**

**NHÓM 7**

**GVHD: ThS. Nguyễn Thị Anh Thư**

**1.**Nguyễn Thế Tiến Đạt **MSSV:** 22520225

**2.**Vũ Quang Huy **MSSV:** 22520587

**3.**Võ Nhất Phương **MSSV:** 22521172

**4.**Lê Nguyễn Thùy Dương **MSSV:** 22520298

**5.**Đoàn Ngọc Thanh Sơn **MSSV:** 21521385

**Tp. Hồ Chí Minh, 03/2024**

**NỘI DUNG THẢO LUẬN**:

**Bài tập: Tìm hiểu các thuật toán theo phương thức học:**

* **Học có giám sát (Supervised Learning)**
* **Học không giám sát (Unsupervised Learning)**
* **Học bán giám sát (Semi-Supervised Learning)**
* **Học tăng cường (Reinforcement Learning)**
* **Học chuyển giao (Transfer Learning)**

**Trả lời:**

**HỌC CÓ GIÁM SÁT (SUPERVISED LEARNING)**

|  | **Neural Networks** | **Naive Bayes** | **Linear Regression** | **Logistic Regression** | |
| --- | --- | --- | --- | --- | --- |
| **Giới thiệu** | Thuật toán xử lý dữ liệu qua việc bắt chước kết nối bộ não con người qua các nodes | Thuật toán được mô hình hóa dựa trên định lý Bayes trong xác suất thống kê | Thuật toán giải thích sự thay đổi một biến phụ thuộc dựa trên sự thay đổi của một biến độc lập | Thuật toán dự đoán xác suất của một biến phụ thuộc nhị phân dựa trên một hoặc nhiều biến độc lập | |
| **Mô hình** | Mô hình thuật toán gồm các input layer, hidden layer và output layer. Mỗi layer sẽ chứa các nodes | - P(y/X): xác suất y khi biết X  - P(X/y): xác suất X khi biết y  - P(y): xác suất y  - P(X): xác suất X | y = B0 + B1\*x1 + \*x2 + … + Bn\*xn + ϵ  - y là biến phụ thuộc cần dự đoán  - x1, x2, xn… là biến độc lập  - B1, B2, Bn.. là hệ số hồi quy  - ϵ là phần dữ liệu không được giải thích, có thể là sai số. | - σ(z): biểu diễn xác suất của biến phụ thuộc nhị phân có giá trị 1  - z: tổng trọng số của các biến độc lập | |
| **Đặc điểm chính** | Mỗi node sẽ chứa một hàm toán học, có chức năng phân loại dữ liệu và đưa đầu ra tương ứng | Có 2 giả thiết đưa ra trong mô hình Naive Bayes:  - Các đặc trưng đưa vào mô hình là độc lập với nhau: sự thay đổi giá trị của một đặc trưng không ảnh hưởng đến các đặc trưng còn lại.  - Các đặc trưng đưa vào mô hình có ảnh hưởng ngang nhau đối với đầu ra mục tiêu. | - Cung cấp cơ sở vững chắc để dự đoán giá trị của một biến dựa trên giá trị của một hoặc nhiều biến khác  - Có thể được áp dụng cho một loạt các vấn đề dự đoán, từ hồi quy tính tuyến đơn giản đến đa biến  - Cho phép phân tích ảnh hưởng của các biến độc lập lên biến phụ thuộc, thông qua các hệ số hồi quy, giúp xác định biến nào có ảnh hưởng lớn nhất đến biến mục tiêu. | - Sử dụng hàm sigmoid (hay còn gọi là hàm logistic) để chuyển đổi các giá trị dự đoán thành xác suất, có thể biến đổi bất kỳ giá trị thực số nào thành một giá trị nằm trong khoảng từ 0 đến 1, làm cho nó phù hợp để diễn giải dưới dạng xác suất.  - Thuật toán dự đoán xác suất mà một quan sát cụ thể thuộc về một lớp nhất định, thay vì chỉ đơn giản phân loại quan sát đó. | |
| **Ưu điểm** | - Có thể phân tích trực quan hiệu quả  - Có thể xử lý dữ liệu phi cấu trúc  - Thay đổi cấu trúc theo mục đích | - Dễ cài đặt  - Có tốc độ xử lý cao | - Trực quan và dễ hiểu  - Không đòi hỏi nhiều tài nguyên tính toán  - Ứng dụng rộng rãi | - Dễ hiểu và giải thích  - Có tính hiệu quả | |
| **Nhược điểm** | - Yêu cầu phần cứng cao  - Kết quả đầu ra có thể thiếu hoặc sai sót  - Phụ thuộc vào dữ liệu đầu vào | - Yêu cầu các đặc trưng đầu vào phải độc lập | - Bị ảnh hưởng bởi nhiều điểm ngoại lệ  - Sai số giữa các quan sát là độc lập và có phân phối chuẩn với phương sai đồng nhất, điều này không phải lúc nào cũng dễ dàng kiểm tra hoặc đạt được trong thực tế | - Chỉ phù hợp với phân loại nhị phân  - Các biến độc lập trong mô hình phải độc lập với nhau | |
| **Ứng dụng** | Máy học, tài chính, kinh doanh, lập kế hoạch mục tiêu, bảo trì sản phẩm, dự báo thời tiết, nghiên cứu tiếp thị, đánh giá rủi ro, phòng chống gian lận,… | Email, Recommendation System, lọc phân loại rác,... | - Dự đoán giá cổ phiếu  - Dự đoán bệnh  - Dự đoán hành vi khách hàng  - Phân tích biến đổi khí hậu  … | - Dự đoán khả năng mắc bệnh dựa trên các yếu tố rủi ro  - Dự đoán hành vi của khách hàng trong marketing  - Phân loại rủi ro tín dụng của khách hàng | |
| **Dữ liệu xử lý** | Dữ liệu qua nhận diện hình ảnh, giọng nói, trong y tế,... | Dữ liệu là xác suất của đối tượng cần phân tích, phân loại xác suất,... | - Biến phụ thuộc: giá cả, doanh số, chỉ số không khí…  - Biến độc lập: thu nhập, số lượng sản phẩm đã bán,... | - Biến phụ thuộc: biến phân loại nhị phân  - Biến độc lập: số liên tục hoặc rời rạc, biến phân loại | |
| **Yêu cầu về hiệu suất máy tính** | Tùy thuộc vào lượng dữ liệu cần xử lý sẽ có yêu cầu khác nhau | Thuật toán này không yêu cầu cao về hiệu suất máy tính mà phụ thuộc vào dữ liệu đầu vào và cách triển khai | Yêu cầu về hiệu suất máy tính thấp do tính toán đơn giản | Yêu cầu về hiệu suất máy tính không cao, nếu lượng dữ liệu xử lý nhiều thì sẽ cần nâng cấp các phần cứng hoặc lưu trữ cloud | |

**HỌC KHÔNG GIÁM SÁT (UNSUPERVISED LEARNING)**

|  | **Clustering** | **Association Rules** | **Dimensionality reduction** |
| --- | --- | --- | --- |
| **Giới thiệu** | Phân cụm các điểm dữ liệu | Dựa trên quy tắc để tìm ra mối quan hệ giữa các biến trong một tập dữ liệu đã cho | Là một kỹ thuật được sử dụng khi số lượng đặc trưng hoặc chiều dữ liệu trong một tập dữ liệu cụ thể quá cao |
| **Loại dữ liệu** | Không có nhãn | Không có nhãn | Không có nhãn |
| **Ứng dụng** | Phân cụm tài liệu, xác định khu vực có nguy cơ phạm tội cao, phân khúc khách hàng, phát hiện gian lận bảo hiểm, phân tích dữ liệu giao thông công cộng, phân cụm cảnh báo IT | Những phương pháp này thường được sử dụng cho phân tích giỏ hàng mua sắm, cho phép các công ty hiểu rõ hơn về mối quan hệ giữa các sản phẩm khác nhau. | Thường được sử dụng trong giai đoạn tiền xử lý dữ liệu, và có một vài phương pháp giảm chiều dữ liệu khác nhau có thể được sử dụng |
| **Ưu điểm** | Dữ liệu có thể được tổ chức thành các nhóm có ý nghĩa sử dụng công cụ phân tích mạnh mẽ của phân tích cụm. Bạn có thể sử dụng nó để xác định các phân đoạn, tìm ra mẫu ẩn, và cải thiện quyết định. | Phân Tích Dữ Liệu Hiệu Quả,Quyết Định Hiệu Quả,Nâng Cao Kinh Nghiệm Khách Hàng, Lợi Thế Cạnh Tranh | Việc giảm số chiều của các đặc trưng đồng nghĩa với việc giảm không gian cần thiết để lưu trữ tập dữ liệu, vì tập dữ liệu cũng bị giảm đi.  Thời gian huấn luyện mô hình ngắn hơn với số chiều giảm.  Việc trực quan hóa dữ liệu nhanh hơn được tạo điều kiện bằng cách giảm số lượng đặc trưng trong tập dữ liệu.  Các đặc trưng lặp lại trong miền đa tuyến đa chiều biến mất. |
| **Nhược điểm** | Nhược điểm của phương pháp phân cụm bao gồm sự nhạy cảm với điều kiện ban đầu, sự phụ thuộc vào việc lựa chọn các tham số, khó khăn trong việc xác định số lượng cụm tối ưu, và thách thức trong việc xử lý dữ liệu nhiều chiều hoặc dữ liệu không hợp ngữ cảnh | Các Phát Hiện Sai  Phạm Vi Hạn Chế  Vấn Đề Về Chất Lượng Dữ Liệu  Vấn Đề Về Chất Lượng Dữ Liệu | Một số dữ liệu có thể bị mất do giảm số chiều.  Trong kỹ thuật giảm chiều PCA, các thành phần chính cần được xem xét đôi khi không được biết trước. |
| **Dữ liệu xử lý** | Dữ liệu số, cụ thể là dữ liệu liên tục, là loại dữ liệu mà trong đó các giá trị có thể nhận một phạm vi không giới hạn các giá trị. | Dữ liệu khách hàng trong quá khứ | Dữ liệu lớn cần lược bỏ |

**HỌC BÁN GIÁM SÁT (SEMI-SUPERVISED LEARNING)**

|  | **Self-training** | **Co-training** | **Label Propagation** |
| --- | --- | --- | --- |
| **Mục đích** | Cải thiện dự đoán bằng cách sử dụng dữ liệu gắn nhãn và không gắn nhãn để đào tạo một mô hình | Tận dụng thông tin từ các tập dữ liệu không gắn nhãn khác nhau để cải thiện hiệu suất | Lan truyền thông tin từ các dữ liệu gắn nhãn đã biết sang dữ liệu không gắn nhãn |
| **Ứng dụng** | Phân loại Email Spam, Nhận dạng Ngôn ngữ | Phân loại Văn bản đa ngôn ngữ, Phân loại Dữ liệu Dòng thời gian | Phát hiện Giao dịch gian lận, Phân cụm Dữ liệu |
| **Đặc điểm chính** | khả năng sử dụng dữ liệu không gắn nhãn để cải thiện hiệu suất của mô hình học máy, đặc biệt là khi tập dữ liệu gắn nhãn có ít mẫu hoặc khi chi phí thu thập dữ liệu gắn nhãn cao. | khả năng sử dụng thông tin từ nhiều nguồn dữ liệu không gắn nhãn để cải thiện hiệu suất của mô hình học máy. Điều này có thể giúp tận dụng hiệu quả thông tin từ các nguồn dữ liệu phong phú và đa dạng, giúp cải thiện khả năng tổng quát hóa của mô hình | khả năng lan truyền thông tin nhãn từ các mẫu gắn nhãn sang các mẫu không gắn nhãn trong tập dữ liệu. Điều này cho phép thuật toán tận dụng thông tin từ các mẫu đã biết để dự đoán nhãn cho các mẫu không gắn nhãn một cách hiệu quả. |
| **Ưu điểm** | Hiệu quả khi có ít dữ liệu gắn nhãn. | Đặc biệt hiệu quả khi dữ liệu không gắn nhãn có cấu trúc đặc biệt. | Cung cấp khả năng mở rộng dữ liệu gắn nhãn từ một số lượng nhỏ dữ liệu gắn nhãn. |
| **Nhược điểm** | Dễ bị ảnh hưởng bởi dữ liệu không gắn nhãn không đại diện. | Đòi hỏi ít dữ liệu gắn nhãn hơn so với self-training | Cần điều chỉnh các tham số cho mỗi tập dữ liệu cụ thể. |
| **Yêu cầu về hiệu suất máy tính** | Thấp hơn so với co-training | Tăng với số lượng mô hình phụ thuộc vào số lượng thuật toán được sử dụng trong quá trình. | Cao hơn so với self-training và co-training do việc tính toán liên quan đến việc lan truyền nhãn |

**HỌC TĂNG CƯỜNG (REINFORCEMENT LEARNING)**

|  | **Q-Learning** | **SARSA (State-Action-Reward-State-Action)** | **Deep Q-Networks (DQN)** |
| --- | --- | --- | --- |
| **Mô hình** | Phương pháp học tăng cường không mô hình (model-free) loại thuật toán off-policy | Phương pháp học tăng cường không mô hình (model-free) loại thuật toán on-policy | Kết hợp Q-Learning với deep learning |
| **Ứng dụng** | Dùng để học một hàm giá trị hành động để tìm hiểu cách hành động trong một môi trường để đạt được mục tiêu cụ thể.   * Ứng dụng trong trò chơi: Q-Learning được sử dụng rộng rãi trong các trò chơi máy tính, như các trò chơi board game, game video, hoặc game trực tuyến. Ví dụ như việc huấn luyện một hệ thống tự học chơi game cờ vây hoặc Pac-Man. * Ứng dụng trong điều khiển tự động: Q-Learning có thể được sử dụng để điều khiển robot hoặc xe tự lái trong một môi trường ảo, giúp chúng tìm ra các hành động tối ưu để đạt được mục tiêu như tránh vật cản hoặc di chuyển đến một điểm đích. | Học trực tiếp chính sách từ dữ liệu, thay vì chỉ tối ưu hóa hàm giá trị hành động.   * Ứng dụng trong robot tự học: SARSA thường được sử dụng trong các ứng dụng robot tự học, nơi robot phải tương tác với môi trường xung quanh và học cách hành động một cách an toàn và hiệu quả. Ví dụ, robot học cách điều khiển cánh tay của mình để chạm vào và nắm bắt các đối tượng một cách an toàn. * Ứng dụng trong hệ thống tự lái: SARSA có thể được áp dụng trong các hệ thống tự lái xe để học cách điều khiển xe trong môi trường giao thông phức tạp, bao gồm quyết định khi nào nên tăng tốc, chuyển làn đường hoặc dừng lại để tránh va chạm. | Sử dụng mạng nơ-ron để ước lượng hàm giá trị hành động trong các không gian trạng thái và hành động lớn.   * Ứng dụng trong game và robot tự học: DQN được sử dụng rộng rãi trong các ứng dụng game và robot tự học, nơi một mạng nơ-ron sâu được sử dụng để ước lượng giá trị hành động tối ưu. Ví dụ, DQN có thể được sử dụng để huấn luyện robot hoặc hệ thống tự học chơi các trò chơi video phức tạp như StarCraft hoặc Dota 2. * Ứng dụng trong hệ thống điều khiển tự động: DQN cũng có thể được sử dụng trong các hệ thống điều khiển tự động như xe tự lái, nơi mạng nơ-ron sâu được sử dụng để dự đoán hành động tối ưu dựa trên dữ liệu từ các cảm biến và môi trường xung quanh. |
| **Đặc điểm chính** | * Q-Learning học từ các cặp trạng thái-hành động và giá trị phần thưởng tương ứng, cố gắng tìm ra hàm giá trị tối ưu cho mỗi hành động trong mỗi trạng thái. * Tại mỗi bước thời gian, thuật toán cập nhật giá trị của hàm Q (giá trị hành động) bằng cách sử dụng các phần thưởng thu được từ hành động thực hiện và giá trị Q của trạng thái tiếp theo. * Q(S,A)=Q(S,A)+a[R+ymaxaQ(S’,a’)-Q(S,A)] | * SARSA cập nhật giá trị của hàm Q bằng cách sử dụng giá trị Q của hành động tiếp theo dựa trên chính sách hiện tại của nó. * Giá trị chênh lệch thời gian được tính bằng cách sử dụng kết hợp hành động trạng thái hiện tại và kết hợp hành động trạng thái tiếp theo. * Q(S,A)=Q(S,A)+a[R+yQ(S,A)-Q(S,A)] | * DQN sử dụng một mạng nơ-ron để ước lượng hàm giá trị hành động (Q-value function) trong các không gian trạng thái và hành động lớn. * DQN học thông qua việc tối ưu hóa một hàm mục tiêu sử dụng phương pháp như gradient descent, nhằm giảm thiểu sai số giữa giá trị ước lượng và giá trị thực tế. |
| **Ưu điểm** | * Không cần biết mô hình (Model-Free): Q-learning không yêu cầu biết trước mô hình của hệ thống. Nó hoạt động dựa trên việc thử nghiệm và cập nhật giá trị Q mà không cần biết chi tiết về quy luật của môi trường. * Tối ưu hóa chính sách tối ưu (Off-Policy): Q-learning là thuật toán off-policy, có nghĩa là nó có thể học từ các hành động không tuân theo chính sách hiện tại. Điều này giúp nó tìm ra chính sách tối ưu mà không bị ảnh hưởng bởi chính sách hiện tại. * Hội tụ đến chính sách tối ưu (Convergence): Với đủ thời gian và điều kiện, Q-learning hội tụ đến chính sách tối ưu, tức là hội tụ đến hàm giá trị Q tốt nhất. * Áp dụng rộng rãi: Q-learning được sử dụng trong nhiều lĩnh vực như trò chơi, robot tự học, quản lý tài nguyên, và nhiều ứng dụng khác. | * On-policy (Tuân theo chính sách hiện tại): SARSA là thuật toán on-policy, có nghĩa là nó học từ các hành động thực tế được thực hiện trong chính sách hiện tại. Điều này giúp nó hội tụ ổn định hơn và không bị ảnh hưởng bởi việc thay đổi chính sách. * Hội tụ ổn định: SARSA hội tụ đến chính sách tối ưu với tốc độ hội tụ ổn định. Nó thường không bị dao động mạnh như Q-learning. * Áp dụng trong các tác vụ tương tác: SARSA thích hợp cho các tác vụ tương tác, nơi tác nhân phải thực hiện hành động và tương tác với môi trường liên tục. * Dễ hiểu và triển khai: Công thức cập nhật của SARSA đơn giản và dễ hiểu, giúp triển khai nhanh chóng. | * Tối ưu hóa giá trị hành động (Q-value): DQN là một phương pháp dựa trên giá trị, giúp tối ưu hóa hàm giá trị Q cho mỗi hành động dựa trên trạng thái hiện tại. Điều này giúp tìm ra chính sách tối ưu mà không cần biết trước mô hình chi tiết của môi trường. * Không cần chờ đến cuối tập (episode): So với thuật toán REINFORCE, DQN không cần chờ đến cuối tập để tính toán phần thưởng cuối cùng. Thay vào đó, nó sử dụng phương trình Bellman để cập nhật giá trị hành động khi di chuyển qua các bước tiếp theo. * Áp dụng cho các môi trường phức tạp: DQN đã chứng minh khả năng học các nhiệm vụ phức tạp trực tiếp từ dữ liệu đầu vào cảm quan, làm cho nó trở thành một bước tiến quan trọng trong lĩnh vực trí tuệ nhân tạo và học tăng cường. * Kết hợp với Double DQN (DDQN): Double DQN (DDQN) là biến thể của DQN giúp giảm hiện tượng đánh giá quá cao của giá trị Q và cải thiện tính ổn định của học tập |
| **Nhược điểm** | * Hội tụ chậm: Q-learning có thể hội tụ chậm đối với một số môi trường phức tạp. Việc cập nhật giá trị Q dựa trên toàn bộ không gian trạng thái và hành động có thể đòi hỏi nhiều thời gian. * Không xử lý được không gian trạng thái lớn: Khi không gian trạng thái quá lớn, việc duyệt và cập nhật giá trị Q trở nên khó khăn và tốn kém về thời gian và tài nguyên. * Dễ bị rơi vào local minimum: Q-learning có thể bị rơi vào local minimum, khi giá trị Q không hội tụ đến chính sách tối ưu mà chỉ tìm được một chính sách tốt nhất trong một phạm vi nhất định. * Không xử lý được môi trường liên tục: Q-learning thường được áp dụng cho môi trường rời rạc. Trong môi trường liên tục, việc xấp xỉ không gian trạng thái và hành động trở nên phức tạp. | * Có thể dẫn đến chậm trong việc hội tụ so với Q-Learning. * Khó mở rộng cho các mô hình lớn. | * Khó khăn trong việc hội tụ: DQN có thể gặp khó khăn trong việc hội tụ đối với một số môi trường phức tạp. Việc xử lý hàm giá trị Q bằng mạng neural có thể dẫn đến hiện tượng dao động và khó kiểm soát. * Cần nhiều dữ liệu huấn luyện: DQN yêu cầu một lượng lớn dữ liệu huấn luyện để học hiệu quả. Điều này đặc biệt đúng khi áp dụng cho các môi trường phức tạp. * Khả năng overfitting: Mạng neural trong DQN có thể dễ dàng bị overfitting, đặc biệt khi không có cơ chế kiểm soát như dropout hoặc regularization. * Khó điều chỉnh các siêu tham số: DQN có nhiều siêu tham số như kích thước batch, tỷ lệ học, số lớp ẩn, và kích thước bộ nhớ đệm. Tinh chỉnh các siêu tham số này đòi hỏi kiến thức và kinh nghiệm. |
| **Dữ liệu xử lý** | Dữ liệu về trạng thái và hành động | Dữ liệu về trạng thái và hành động | Dữ liệu lớn, phức tạp |
| **Yêu cầu về hiệu suất máy tính** | Tùy thuộc vào kích thước của không gian trạng thái và hành động | Tương đối cao | Cao, đặc biệt là trong quá trình huấn luyện |

**HỌC CHUYỂN GIAO (DEEP TRANSFER LEARNING)**

|  | **Transfer learning to same task(TLST)** | **Transfer Learning to New Task (TLNT)** | **Multi-Source Transfer Learning (MSTL)** |
| --- | --- | --- | --- |
| **Giới thiệu** | Trong loại này, mô hình được huấn luyện trước đó trên một tập dữ liệu lớn, và sau đó được chuyển giao trực tiếp cho cùng một tác vụ, hoặc tác vụ rất giống với tác vụ gốc. Ví dụ, một mô hình được huấn luyện trên tập dữ liệu lớn về nhận diện khuôn mặt có thể được chuyển giao trực tiếp cho việc nhận diện biểu cảm khuôn mặt. | Loại này thường liên quan đến việc chuyển đổi mô hình đã được huấn luyện trước đó trên một nhiệm vụ khác cho một tác vụ mới mà không giống với bất kỳ tác vụ nào mà mô hình ban đầu đã được huấn luyện. Ví dụ, một mô hình được huấn luyện trên tập dữ liệu về nhận diện đối tượng trong ảnh có thể được chuyển giao cho việc nhận diện bệnh. | Trong loại này, kiến thức từ nhiều nguồn khác nhau được kết hợp và chuyển giao sang một tác vụ mới. Điều này có thể bao gồm việc sử dụng kiến thức từ nhiều mô hình đã được huấn luyện trước đó hoặc sử dụng dữ liệu từ nhiều nguồn khác nhau |
| **Đặc điểm chính** | * Trong TLST, mô hình được chuyển giao từ một tác vụ gốc tương tự hoặc cùng một tác vụ cho tác vụ mới. * Dữ liệu và đặc trưng giữa tác vụ gốc và tác vụ mới có thể tương tự hoặc có sự khác biệt nhỏ. | * Trong TLNT, mô hình được chuyển giao từ một tác vụ không liên quan hoặc tương tự nhỏ đến một tác vụ mới. * Dữ liệu và đặc trưng giữa tác vụ gốc và tác vụ mới thường có sự khác biệt đáng kể. | * Trong MSTL, kiến thức từ nhiều nguồn khác nhau được kết hợp và chuyển giao sang một tác vụ mới. * Dữ liệu và đặc trưng từ các nguồn khác nhau có thể có tính tương đồng hoặc đa dạng. |
| **Ưu điểm** | * Tiết kiệm thời gian và tài nguyên: Không cần phải huấn luyện mô hình từ đầu, giúp tiết kiệm thời gian và công sức đào tạo. * Hiệu suất khá: Nếu tác vụ mới tương tự hoặc giống với tác vụ gốc, TLST thường mang lại hiệu suất tốt trên tác vụ mới. * Hiệu quả về chi phí: Sử dụng lại mô hình đã huấn luyện trước đó giúp giảm chi phí đầu tư cho việc huấn luyện. | * Mở rộng khả năng ứng dụng: TLNT cho phép chuyển đổi kiến thức từ các tác vụ không liên quan thành các tác vụ mới, mở rộng khả năng ứng dụng của mô hình. * Tính linh hoạt: Có thể áp dụng cho nhiều tác vụ mới mà không cần phải huấn luyện lại mô hình từ đầu. | * Tận dụng đa dạng kiến thức: MSTL cho phép kết hợp và tận dụng kiến thức từ nhiều nguồn khác nhau, giúp cải thiện hiệu suất của mô hình trên tác vụ mới. * Giảm overfitting: Bằng cách kết hợp kiến thức từ nhiều nguồn, MSTL có thể giảm nguy cơ overfitting và cải thiện khả năng tổng quát hóa của mô hình. |
| **Nhược điểm** | * Hạn chế của kiến thức chuyển giao: Nếu có sự khác biệt lớn giữa tác vụ gốc và tác vụ mới, hiệu suất của mô hình có thể bị ảnh hưởng. * Nguy cơ overfitting: Nếu không kiểm soát cẩn thận, có thể xảy ra hiện tượng overfitting trên dữ liệu mới. | * Khó khăn trong chuyển đổi kiến thức: Có thể gặp khó khăn trong việc chuyển đổi kiến thức từ các tác vụ không liên quan. * Yêu cầu dữ liệu đào tạo lớn: Đôi khi cần một lượng lớn dữ liệu đào tạo mới để hiệu suất của mô hình đạt được mong muốn. | * Phức tạp hơn: MSTL thường đòi hỏi các phương pháp và thuật toán phức tạp hơn so với TLST và TLNT. * Yêu cầu tính linh hoạt của mô hình: Để kết hợp kiến thức từ nhiều nguồn, mô hình phải có tính linh hoạt cao và khả năng học được từ nhiều nguồn khác nhau. |
| **Ứng dụng** | * Fine-tuning: Một trong những phương pháp phổ biến trong TLST là fine-tuning, trong đó mô hình được huấn luyện tiếp trên tập dữ liệu mới với một tốc độ học thấp để cập nhật các trọng số của mạng nơ-ron. * Feature extraction: Một phương pháp khác là sử dụng mạng nơ-ron đã được huấn luyện trước đó để trích xuất các đặc trưng từ dữ liệu mới, sau đó sử dụng các đặc trưng này để huấn luyện một mô hình mới. | * Fine-tuning: Một trong những phương pháp phổ biến trong TLNT là fine-tuning, trong đó mô hình được huấn luyện tiếp trên tập dữ liệu mới với một tốc độ học thấp để cập nhật các trọng số của mạng nơ-ron. * Transfer learning layer: Một phương pháp khác là sử dụng một phần của mô hình nguồn như một feature extractor và thêm một số lớp mới vào mô hình để phù hợp với tác vụ mới. | * Merging features: Kết hợp các đặc trưng được trích xuất từ nhiều nguồn dữ liệu thành một tập đặc trưng đồng nhất trước khi áp dụng vào mô hình. * Adaptive weighting: Sử dụng trọng số động để đánh giá độ quan trọng của mỗi nguồn dữ |
| **Yêu cầu về hiệu suất** | * Trong TLST, thường chỉ cần thực hiện fine-tuning trên một phần nhỏ của mô hình. Do đó, cần ít tài nguyên tính toán hơn so với việc huấn luyện ban đầu. | * Việc triển khai TLNT có thể đòi hỏi máy tính có khả năng xử lý cao, đặc biệt là khi cần thực hiện huấn luyện mô hình từ đầu. | * MSTL thường đòi hỏi máy tính có khả năng xử lý linh hoạt để có thể kết hợp kiến thức từ nhiều nguồn khác nhau. |