[Bài 6: Convolutional neural network | Deep Learning cơ bản (nttuan8.com)](https://nttuan8.com/bai-6-convolutional-neural-network/)

Convolutional Neural Networks (CNNs), là kiến trúc mạng neural nhân tạo nâng cao, được xây dựng để giải quyết các bài toán phức tạp, đặc biệt là các bài toán liên quan đến xử lý hình ảnh, chẳng hạn như nhận diện đối tượng, phân loại hình ảnh, thậm chí là tự động lái xe.

Cấu trúc cơ bản của CNNs gồm các lớp sau:

1. Lớp tích chập (Convolutional layer): Lớp này sử dụng các bộ lọc (kernels) để thực hiện các phép tích chập dựa trên ảnh đầu vào để cho ra tín hiệu mới giảm bớt các đặc trưng mà không cần thiết, giữ lại các đặc trưng chính và quan trọng. Để dễ hiểu hơn, ta có thể coi nó như là một cửa sổ trượt dưới một ma trận, và kết quả của 2 phép tích chập ma trận là một ma trận mới chứa các đặc trưng của ma trận đầu vào như ví dụ bên dưới:

Ta định nghĩa kernel là một ma trận vuông kích thước k\*k trong đó k là số lẻ. k có thể bằng 1, 3, 5, 7, 9,… Ví dụ kernel kích thước 3\*3

A number and a line

Description automatically generated with medium confidence

Với mỗi phần tử xij trong ma trận X lấy ra một ma trận có kích thước bằng kích thước của kernel W có phần tử xij làm trung tâm (đây là vì sao kích thước của kernel thường lẻ) gọi là ma trận A. Sau đó tính tổng các phần tử của phép tính element-wise của ma trận A và ma trận W, rồi viết vào ma trận kết quả Y.

A grid with numbers and symbols

Description automatically generated

Để tính giá trị của x22 ta nhân từng giá trị của ma trận A (được tô màu cam) với ma trận W, kết quả được ma trận Y. Tương tự với các giá trị khác để ra đưuọc kế quả bên dưới:

A screenshot of a grid

Description automatically generated

Vậy với các giá trị viền ngoài như x11 thì không thể tính được, vậy nên ta them một lớp bọc bên ngoài gọi là padding để giải quyết vấn đề trên.

A grid with numbers and a red square

Description automatically generated

Mục đích của phép tính convolution trên ảnh là làm mở, làm nét ảnh; xác định các đường;… Mỗi kernel khác nhau thì sẽ phép tính convolution sẽ có ý nghĩa khác nhau. Vì vậy tùy từng bài toán hoặc từng mục đích mà chúng ta chọn kernel phù hợp:

A screenshot of a computer program

Description automatically generated

Đối với các ảnh màu có tới 3 kênh là red, green, blue thì ta phải thực hiện phép tích chập bằng cách nhân một tensor 3 chiều với một kernel 3 chiều, một phần quan trọng của quá trình này là việc thêm một số gọi là "bias" sau khi thực hiện phép nhân và tổng hợp.

Bias là một giá trị số được thêm vào sau khi tích chập để tạo ra tính phi tuyến tính trong mạng neural. Nó cho phép mô hình học và biểu diễn các đặc trưng tại các vị trí không phụ thuộc vào dữ liệu đầu vào. Trong trường hợp mạng tích chập, mỗi kênh đầu ra của tích chập sẽ có một bias riêng. Chúng ta chỉ cần khởi tạo bias và giá trị bias sẽ được thực hiện tự động bởi các thư viện và framework như TensorFlow, PyTorch, Keras, …

A diagram of a program

Description automatically generated with medium confidence

Tổng quát convolution layer:

A white background with black text and red text

Description automatically generated

Output của convolutional layer sẽ qua hàm activation function trước khi trở thành input của convolutional layer tiếp theo.

Tổng số parameter của layer: Mỗi kernel có kích thước F\*F\*D và có 1 hệ số bias, nên tổng parameter của 1 kernel là F\*F\*D + 1. Mà convolutional layer áp dụng K kernel => Tổng số parameter trong layer này là K \* (F\*F\*D + 1).

1. Lớp tổng hợp (Pooling layer): Lớp tổng hợp giúp giảm kích thước khối ma trận đầu vào thông qua việc tìm ra 1 giá trị đại diện cho mỗi vùng không gian mà bộ lọc đi qua, không làm thay đổi đường nét chính của bức ảnh nhưng giảm được kích thước. Điều này giúp giảm độ phức tạp của mô hình và giúp tránh overfitting.

Gọi pooling size kích thước K\*K. Input của pooling layer có kích thước H\*W\*D, ta tách ra làm D ma trận kích thước H\*W. Với mỗi ma trận, trên vùng kích thước K\*K trên ma trận ta tìm maximum hoặc average của dữ liệu rồi viết vào ma trận kết quả.

A close-up of a grid

Description automatically generated

Có 2 loại pooling layer phổ biến là: max pooling và average pooling.

A diagram of a number

Description automatically generated

1. Lớp kích hoạt (Activation layer): Các lớp này thường được sử dụng hàm kích hoạt như ReLU (Rectified Linear Unit) để thêm tính phi tuyến tính vào mô hình
2. Lớp kết nối đầy đủ (Fully Connected Layer): Sau khi qua các lớp tích chập và tổng hợp, các đặc trưng được duyệt qua các lớp kết nối đầy đủ để thực hiện phân loại hoặc dự đoán.

A screenshot of a computer

Description automatically generated

Tổng quát quá trình trong CNNs:

A diagram of a rectangular object

Description automatically generated

Ưu điểm của CNNs

CNNs có nhiều ưu điểm quan trọng, bao gồm:

Khả năng học đặc trưng tự động: CNNs tự động học các đặc trưng quan trọng từ dữ liệu, giúp giảm bớt sự phụ thuộc vào việc tạo ra các đặc trưng thủ công.

Invariance về dịch chuyển: CNNs có khả năng nhận diện đối tượng trong ảnh bất kể vị trí của chúng, điều này có ích trong nhiều ứng dụng thực tế.

Hiệu suất cao trên dữ liệu hình ảnh: CNNs đã đạt được hiệu suất xuất sắc trong nhiều cuộc thi và ứng dụng thực tế, chẳng hạn như phát hiện khuôn mặt, phân loại hình ảnh và xe tự động lái.

Ứng dụng của CNNs

CNNs có nhiều ứng dụng quan trọng, bao gồm:

Nhận diện đối tượng: CNNs có thể được sử dụng để nhận diện đối tượng trong ảnh và video, từ nhận dạng khuôn mặt đến phát hiện xe hơi.

Phân loại hình ảnh: CNNs có thể phân loại ảnh vào các lớp khác nhau, ví dụ như phân loại loại động vật hoặc thực phẩm.

<them hinh anh>

Xử lý ảnh y tế: Trong lĩnh vực y tế, CNNs đã được sử dụng để xác định bệnh từ hình ảnh chẩn đoán như tia X hoặc hình ảnh MRI.

Xe tự động lái: CNNs đóng một vai trò quan trọng trong phát triển công nghệ xe tự động lái, giúp xe ô tô tự động nhận biết và phản ứng với môi trường xung quanh.

Một số hang nổi tiếng đang sử dụng CNNs như: Apple đã sử dụng CNNs trong Face ID, công nghệ nhận dạng khuôn mặt trên các phiên bản Iphone mới nhất. Google sử dụng CNNs vào Google Photos để phân loại và tìm kiếm hình ảnh dựa trên nội dung, và cũng đang đưuọc áp dụng vào Waymo, công ty con của Alphabet để phát triển công nghệ xe tự động lái. Facebook cũng sử dụng CNNs để nhận dạng khuôn mặt và phân loại hình ảnh trên nền tảng của họ. NVIDIA công ty chuyên về đồ họa cũng sử dụng CNNs cho mục đích xử lý hình ảnh và thời gian thực bao gồm các giải pháp tự động lái xe.

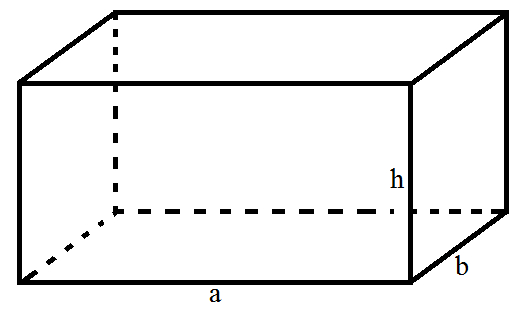
Tensor là gì

Khi dữ liệu biểu diễn dạng 1 chiều, người ta gọi là vector, mặc định khi viết vector sẽ viết dưới dạng cột.

Khi dữ liệu dạng 2 chiều, người ta gọi là ma trận, kích thước là số hàng \* số cột.

Khi dữ liệu nhiều hơn 2 nhiều thì sẽ được gọi là tensor, ví dụ như dữ liệu có 3 chiều.

Để ý thì thấy là ma trận là sự kết hợp của các vector cùng kích thước. Xếp n vector kích thước m cạnh nhau thì sẽ được ma trận m\*n. Thì tensor 3 chiều cũng là sự kết hợp của các ma trận cùng kích thước, xếp k ma trận kích thước m\*n lên nhau sẽ được tensor kích thước m\*n\*k.



AUTOENCODER

Mục đích của autoencoder là tạo ra giá trị input gần đúng bằng cách chỉ tập trung vào các đặc trưng cần thiết. Trong thực tế, một autoencoder là một tập hợp các hạn chế buộc mạng phải tìm hiểu những cách thức mới để đại diện cho dữ liệu, khác với chỉ đơn thuần là sao chép đầu ra.

Một autoencoder điển hình được xác định với một đầu vào, một đại diện bên trong và một đầu ra (đầu ra này xấp xỉ của đầu vào). Việc học tập xảy ra trong các layers gắn với biểu diễn bên trong.

A diagram of a diagram

Description automatically generated

Autoencoders là một loại mạng neural sâu được sử dụng trong học không giám sát để học biểu diễn hiệu quả của dữ liệu. Autoencoders thường được sử dụng trong nhiều tác vụ như phân tích phân loại, nén dữ liệu và tái tạo hình ảnh. Chúng hoạt động bằng cách ánh xạ dữ liệu từ không gian đầu vào sang không gian ẩn và sau đó tái tạo lại dữ liệu từ không gian ẩn

Một autoencoder cơ bản bao gồm hai phần chính:

Encoder (Bộ mã hóa): Phần này ánh xạ dữ liệu đầu vào vào không gian ẩn (latent space). Nó thường bao gồm một hoặc nhiều lớp neural để giảm chiều dữ liệu và biểu diễn nó trong không gian ẩn. Quá trình này tương tự như việc nén dữ liệu.

S= E(x)

Trong đó x là input Data, E là encoder, và s là output trong không gian ẩn

Decoder (Bộ giải mã): Phần này ánh xạ từ không gian ẩn trở lại không gian đầu vào. Nó có nhiệm vụ tái tạo dữ liệu từ biểu diễn ẩn. Decoder thường chứa các lớp neural để phục hồi dữ liệu ban đầu từ không gian ẩn.

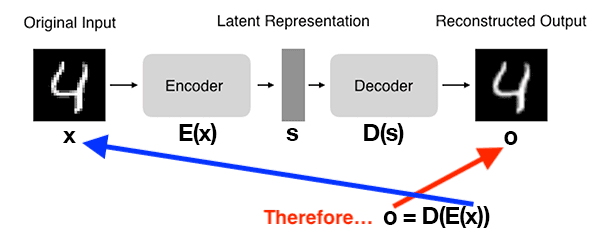
O= D(s)

Trong đó o là output cho Decoder D

Công thức chung cho toàn bộ quá trình Autoencoder là:

O=D(E(x))

Được minh họa như sau:



Trên thực tế, có hai khối layers chính trông giống như một mạng neural truyền thống. Sự khác biệt nhỏ là layer chứa đầu ra phải bằng với đầu vào. Trong hình bên dưới, đầu vào ban đầu đi vào khối đầu tiên được gọi là encoder. Biểu diễn bên trong này nén (giảm) kích thước của đầu vào. Trong khối thứ hai xảy ra việc tái tạo đầu vào. Đây là giai đoạn giải mã.

A diagram of a machine learning

Description automatically generated

Autoencoder cố gắng học biểu diễn sao cho đầu ra của quá trình giải mã gần giống với dữ liệu đầu vào ban đầu. Điều này thúc đẩy mô hình học cách biểu diễn dữ liệu một cách hiệu quả bằng cách tìm ra các đặc trưng quan trọng và loại bỏ nhiễu.

Ứng dụng của autoencoders bao gồm:

Nén dữ liệu: Autoencoders có thể được sử dụng để nén dữ liệu mà vẫn giữ lại thông tin quan trọng. Ví dụ, trong nén hình ảnh để giảm dung lượng lưu trữ.

Phát hiện nhiễu và ngoại lai: Autoencoders có khả năng phát hiện nhiễu hoặc các điểm dữ liệu ngoại lai bằng cách so sánh đầu ra của mô hình với dữ liệu ban đầu.

Trích xuất đặc trưng: Autoencoders có thể được sử dụng để trích xuất các đặc trưng quan trọng từ dữ liệu, chẳng hạn như trong việc phân loại hình ảnh hoặc âm thanh.

Autoencoders là một phần quan trọng của Deep Learning và có nhiều biến thể như Variational Autoencoders (VAEs) và Denoising Autoencoders (DAEs), mở ra nhiều cơ hội cho nghiên cứu và ứng dụng trong nhiều lĩnh vực khác nhau.

MACHINE LEARNING

Machine learning là một nhánh của trí tuệ nhân tạo (AI) và khoa học máy tính, tập trung vào việc sử dụng dữ liệu và thuật toán để bắt chước hành động của con người, dần dần cải thiện độ chính xác của nó.

A robot holding a tablet

Description automatically generated

Machine learning còn là một thành phần quan trọng của lĩnh vực khoa học dữ liệu đang phát triển. Thông qua việc sử dụng các phương pháp thống kê, các thuật toán được đào tạo để đưa ra các phân loại hoặc dự đoán và khám phá những thông tin chi tiết từ chính các dự án khai thác dữ liệu.

Thông qua các thông tin chi tiết có được để thúc đẩy việc đưa ra quyết định đối với các ứng dụng và doanh nghiệp, tác động mạnh đến các chỉ số tăng trưởng. Khi dữ liệu lớn tiếp tục nhu cầu mở rộng và phát triển đòi hỏi nhu cầu tuyển dụng các nhà khoa học dữ liệu sẽ tăng lên. Họ sẽ được yêu cầu giúp xác định các câu hỏi kinh doanh có liên quan nhất và dữ liệu để trả lời chúng.

Bài toán của machine learning thường được chia làm hai loại là dự đoán (prediction) và phân loại (classification). Các bài toán dự đoán thường là giá nhà, giá xe, v.v, còn các bài toán phân loại thường là nhận diện chữ viết tay, đồ vật, v.v.

Machine Learning bao gồm các khái niệm và thành phần quan trọng sau:

Dữ liệu (Data): Dữ liệu là nguyên liệu cơ bản cho quá trình học máy. Nó có thể là dữ liệu số học, hình ảnh, văn bản, âm thanh, hoặc bất kỳ loại dữ liệu nào mà mô hình cần để học và dự đoán.

Mô hình (Model): Mô hình là một thuật toán hoặc mạng neural được thiết kế để học từ dữ liệu và thực hiện nhiệm vụ cụ thể, chẳng hạn như phân loại, dự đoán, hoặc gợi ý. Các mô hình có thể được đào tạo và điều chỉnh để cải thiện hiệu suất.

Học tập (Learning): Quá trình học máy là quá trình mô hình hóa mối quan hệ từ dữ liệu. Học tập có thể là học có giám sát, học không giám sát hoặc học tăng cường, tùy thuộc vào loại nhiệm vụ và dữ liệu.

Áp dụng (Inference): Sau khi mô hình đã được đào tạo, nó có thể được sử dụng để thực hiện các dự đoán hoặc phân loại trên dữ liệu mới mà nó chưa từng thấy trước đó.

Có ba loại học tập chính của Machine Learning:

Học có giám sát (Supervised Learning): Trong loại này, mô hình được đào tạo trên một tập dữ liệu có nhãn, nghĩa là mỗi mẫu dữ liệu đã được gán một nhãn hoặc đầu ra mong muốn. Mô hình học cách ánh xạ đầu vào vào đầu ra dự đoán dựa trên mối quan hệ đã biết từ dữ liệu huấn luyện. Supervised learning còn được gọi là máy học có giám sát, được định nghĩa bằng cách sử dụng các tập dữ liệu được gắn nhãn để huấn luyện các thuật toán phân loại dữ liệu hoặc dự đoán kết quả một cách chính xác.

Khi dữ liệu đầu vào được đưa vào mô hình, mô hình sẽ điều chỉnh trọng lượng của nó cho đến khi nó được lắp một cách thích hợp. Điều này xảy ra như một phần của quá trình xác nhận chéo để đảm bảo rằng mô hình tránh trang bị quá nhiều hoặc trang bị thiếu thông tin.

Supervised machine learning giúp các tổ chức giải quyết nhiều vấn đề trong thế giới thực trên quy mô lớn, chẳng hạn như phân loại thư rác trong một thư mục riêng biệt từ hộp thư đến của bạn.

Một số phương pháp được sử dụng trong Supervised machine learning bao gồm: logistic regression, neural networks, linear regression, naive bayes, random forest, và support vector machine (SVM)

Học không giám sát (Unsupervised Learning): Trong loại này, mô hình được đào tạo trên một tập dữ liệu không có nhãn. Mô hình cố gắng tìm ra cấu trúc hoặc nhóm dữ liệu tự nhiên trong tập dữ liệu mà không cần biết trước thông tin về đầu ra.

Học tăng cường (Reinforcement Learning): Loại này thường được sử dụng trong việc đào tạo máy tính để thực hiện các tác vụ trong môi trường động. Mô hình tương tác với môi trường, nhận được phản hồi từ môi trường về hành động của nó, và điều chỉnh hành vi của mình để tối ưu hóa mục tiêu cụ thể.

Ứng dụng của Machine Learning

Machine Learning đã có sự ảnh hưởng lớn đến nhiều lĩnh vực và ngành công nghiệp, bao gồm:

Thị trường tài chính: Dự đoán giá cổ phiếu, phát hiện gian lận tín dụng, và tối ưu hóa quản lý rủi ro.

Y tế: Phát hiện bệnh, dự đoán kết quả điều trị, và phân tích hình ảnh y tế.

Công nghiệp sản xuất: Tối ưu hóa dây chuyền sản xuất, dự đoán thời gian bảo dưỡng máy móc, và kiểm tra chất lượng sản phẩm.

Ngôn ngữ tự nhiên: Xây dựng chatbot và hệ thống dịch thuật tự động.

Tự động hóa và xe tự hành: Phát triển xe tự lái và robot tự động hoá.

Machine Learning đang phát triển nhanh chóng và có tiềm năng lớn trong tương lai. Các tiến bộ trong deep learning và việc sử dụng dữ liệu lớn (big data) đang mở ra nhiều cơ hội mới. Tuy nhiên, cũng cần quan tâm đến vấn đề về đạo đức và quản lý dữ liệu trong quá trình sử dụng ML. Sự phát triển của ML sẽ tiếp tục làm thay đổi thế giới của chúng ta và tạo ra những cơ hội mới trong tương lai.

DEEP LEARNING

Deep Learning (DL), hay Học Sâu trong tiếng Việt, là một lĩnh vực quan trọng trong Machine Learning (ML) và trí tuệ nhân tạo (Artificial Intelligence - AI) mà nó tập trung vào việc sử dụng các mạng neural sâu để học và biểu diễn dữ liệu phức tạp. DL đã đánh bại nhiều kỷ lục trong các nhiệm vụ như nhận diện hình ảnh, xử lý ngôn ngữ tự nhiên, và nhiều lĩnh vực khác. Deep Learning thúc đẩy nhiều ứng dụng và dịch vụ trí tuệ nhân tạo (AI) nhằm cải thiện tự động hóa, thực hiện các tác vụ phân tích và vật lý mà không cần sự can thiệp của con người.

Deep Learning bao gồm các yếu tố quan trọng sau:

Mạng Neural Sâu (Deep Neural Networks): Điểm nổi bật của Deep Learning là việc sử dụng các mạng neural sâu, có nhiều lớp (layers) neural chồng lên nhau. Các mạng neural sâu này có khả năng học cách biểu diễn dữ liệu ở nhiều mức độ trừu tượng, từ các đặc trưng cơ bản đến các đặc trưng phức tạp hơn.

Học Tự Động (Feature Learning): Deep Learning cho phép mô hình tự động học cách biểu diễn dữ liệu bằng cách trích xuất các đặc trưng quan trọng từ dữ liệu đầu vào. Quá trình này gắn liền với khả năng tự tối ưu hóa của mô hình.

Dữ liệu lớn (Big Data): Deep Learning thường yêu cầu dữ liệu lớn để đào tạo mạng neural hiệu quả. Dữ liệu lớn giúp mô hình học được biểu diễn tốt hơn và đạt hiệu suất cao.

Công nghệ deep learning được ứng dụng rộng rãi trong các sản phẩm và dịch vụ hàng ngày chẳng hạn như trợ lý kỹ thuật số, điều khiển từ xa hỗ trợ giọng nói và phát hiện gian lận thẻ tín dụng cũng như các công nghệ mới nổi chẳng hạn như ô tô tự lái.



Học sâu deep learning hoạt động bằng cách khám phá các cấu trúc phức tạp trong dữ liệu mà chúng trải nghiệm. Cụ thể là xây dựng các mô hình tính toán bao gồm nhiều lớp xử lý, mạng có thể tạo ra nhiều mức trừu tượng để biểu diễn dữ liệu.

Ví dụ, một mô hình học sâu được gọi là mạng nơ-ron phức hợp có thể được đào tạo bằng cách sử dụng số lượng lớn (hàng triệu) hình ảnh, chẳng hạn như những hình ảnh có chứa mèo.

Loại mạng nơ-ron này thường học hỏi từ các pixel có trong hình ảnh mà nó thu được. Nó có thể phân loại các nhóm pixel đại diện cho các đặc điểm của mèo, với các nhóm đặc điểm như móng vuốt, tai và mắt cho biết sự hiện diện của mèo trong hình ảnh.

Đối với nhiều tác vụ, chẳng hạn như thị giác máy tính, nhận dạng giọng nói còn được gọi là xử lý ngôn ngữ tự nhiên, dịch máy và robot, hiệu suất của các hệ thống deep learning vượt xa so với các hệ thống machine learning.

Điều này không có nghĩa là việc xây dựng các hệ thống học sâu là tương đối dễ dàng so với các hệ thống học máy thông thường. Mặc dù nhận dạng tính năng tự trị trong học sâu, hàng nghìn siêu tham số (nút bấm) cần được điều chỉnh để mô hình học sâu deep learning trở nên hiệu quả.

Để hiểu rõ về Deep Learning, bạn cần nắm vững các khái niệm sau:

Kiến thức về Mạng Neural (Neural Networks): Hiểu cách hoạt động của các lớp neural, kích thước của các lớp, và hàm kích hoạt.

Hàm Mất Mát (Loss Function): Hiểu cách thiết lập hàm mất mát phù hợp cho nhiệm vụ cụ thể và cách mô hình cố gắng tối ưu hóa hàm này.

Các Phương pháp Đào Tạo (Training Methods): Hiểu về quá trình lan truyền ngược (Backpropagation) và thuật toán tối ưu hóa như Gradient Descent.

Regularization: Hiểu về các phương pháp như dropout và L2 regularization để tránh overfitting.

Các kiến thức về kiến thức về Convolutional Neural Networks (CNNs): Đối với xử lý hình ảnh, hiểu cách CNNs hoạt động và cách chúng trích xuất đặc trưng từ hình ảnh.

Các kiến thức về Recurrent Neural Networks (RNNs): Đối với xử lý dữ liệu chuỗi (sequence data), hiểu cách RNNs hoạt động và ứng dụng của chúng trong việc mô hình chuỗi dữ liệu

Deep Learning có nhiều ứng dụng quan trọng, bao gồm:

Nhận diện hình ảnh: Deep Learning đã giúp nâng cao hiệu suất trong việc nhận diện đối tượng và khuôn mặt trong hình ảnh.

Xử lý ngôn ngữ tự nhiên (NLP): Deep Learning đã cải thiện hiệu suất trong các nhiệm vụ như dự đoán từ vựng, dịch máy, và phân tích cảm xúc từ văn bản.

Tự động hóa và robot tự động hoá: Deep Learning được sử dụng trong xe tự lái, robot tự động hoá và quản lý dây chuyền sản xuất.

Y tế: Deep Learning được sử dụng trong việc dự đoán bệnh, phát hiện tế bào ác tính, và phân tích hình ảnh y tế.

Deep Learning vẫn đang phát triển và có tiềm năng lớn. Các nghiên cứu mới về mạng neural sâu và học tăng cường đang mở ra nhiều cơ hội mới. Tuy nhiên, cũng cần quan tâm đến vấn đề về đạo đức và đảm bảo tính riêng tư trong việc sử dụng DL.

Deep Learning đã thay đổi cách chúng ta xử lý và hiểu dữ liệu phức tạp. Nó đóng vai trò quan trọng trong trí tuệ nhân tạo và có ứng dụng rộng rãi trong nhiều lĩnh vực. Sự phát triển của DL sẽ tiếp tục thúc đẩy sự tiến bộ và tạo ra những cơ hội mới trong tương lai.

TRANSFORM TRONG DEEP LEARNING

Transform trong Deep Learning là một khái niệm quan trọng và phổ biến, đóng vai trò quan trọng trong việc biến đổi dữ liệu đầu vào hoặc biểu diễn của dữ liệu để cải thiện khả năng học của mô hình. Trong báo cáo này, chúng ta sẽ xem xét các khía cạnh khác nhau của transform trong Deep Learning, bao gồm các phương pháp biến đổi dữ liệu và biến đổi không gian.

Các phương pháp biến đổi dữ liệu trong transform là:

1. Chuẩn hóa (Normalization)

Chuẩn hóa là một biến đổi cơ bản trong Deep Learning. Nó bao gồm việc đưa dữ liệu về miền giá trị cụ thể, thường là giữa 0 và 1 hoặc -1 và 1. Điều này giúp cân bằng tỷ lệ giữa các đặc trưng và giúp mô hình học nhanh hơn. Ví dụ, trong mạng neural, chuẩn hóa có thể thực hiện thông qua việc trừ đi giá trị trung bình và chia cho độ lệch chuẩn của dữ liệu.

2. Data Augmentation

Data Augmentation là một phương pháp mạnh mẽ trong xử lý hình ảnh. Nó bao gồm tạo ra các biến thể của dữ liệu bằng cách thực hiện các phép biến đổi như xoay, thu phóng, cắt, hoặc thay đổi ánh sáng. Việc này không chỉ tạo ra thêm dữ liệu huấn luyện, mà còn giúp ngăn chặn overfitting và cải thiện khả năng tổng quát hóa của mô hình.

3. Biến đổi phi tuyến tính (Non-linear Transformations)

Biến đổi phi tuyến tính thường được sử dụng để biến đổi đặc trưng của dữ liệu thành các biểu diễn phức tạp hơn. Ví dụ, việc áp dụng hàm kích hoạt phi tuyến tính như hàm ReLU (Rectified Linear Unit) có thể giúp mô hình học được biểu diễn phi tuyến tính của dữ liệu.

III. Biến Đổi Không Gian (Space Transformations)

1. Convolution (Tích chập)

Convolution là một biến đổi quan trọng trong xử lý hình ảnh và xử lý chuỗi. Nó được sử dụng để trích xuất các đặc trưng từ dữ liệu bằng cách áp dụng một bộ lọc (kernel) trượt qua dữ liệu. Việc này giúp mô hình nhận biết các đặc điểm cục bộ trong dữ liệu, chẳng hạn như cạnh, góc, hoặc mẫu chuỗi.

2. Pooling (Tích hợp)

Pooling là một phép biến đổi không gian thường được sử dụng sau Convolution. Nó giúp giảm kích thước của biểu diễn không gian, giúp giảm chi phí tính toán và tránh overfitting. Phép tích hợp thông thường bao gồm việc lấy giá trị lớn nhất (Max Pooling) hoặc lấy giá trị trung bình (Average Pooling) trong một vùng cụ thể của dữ liệu.

IV. Ứng Dụng Của Transform Trong Deep Learning

Transform đóng vai trò quan trọng trong nhiều lĩnh vực và nhiệm vụ Deep Learning, bao gồm:

Xử lý hình ảnh: Chuẩn hóa, Data Augmentation, Convolution và Pooling giúp cải thiện hiệu suất mô hình hình ảnh nhận diện và phân loại.

Xử lý ngôn ngữ tự nhiên (NLP): Biến đổi phi tuyến tính và chuẩn hóa được sử dụng để xử lý văn bản và biểu diễn các từ và câu

Tự động hóa và robot tự động hoá: Biến đổi không gian như tích chập và pooling được sử dụng trong việc xử lý dữ liệu cảm giác và điều khiển robot.

V. Kết Luận

Transform trong Deep Learning là một khía cạnh quan trọng giúp cải thiện hiệu suất của mô hình bằng cách biến đổi dữ liệu hoặc biểu diễn của dữ liệu. Hiểu và sử dụng đúng các phương pháp biến đổi có thể giúp tối ưu hóa mô hình và cải thiện khả năng tổng quát hóa của nó trong nhiều nhiệm vụ khác nhau.