**Graph Neural Network**

# 1. Neural Network.

Neural Network là một chuỗi những thuật toán được đưa ra để tìm kiếm các mối quan hệ cơ bản trong tập hợp các dữ liệu. Thông qua việc bắt bước cách thức hoạt động từ não bộ con người.

Neural Network có sự tương đồng chuẩn mạnh với những phương pháp thống kê như đồ thị đường cong và phân tích hồi quy. Neural Network có chứa những lớp bao hàm các nút được liên kết lại với nhau. Mỗi nút lại là một tri giác có cấu tạo tương tự với **hàm hồi quy đa tuyến tính**. Bên trong một lớp, chúng sẽ được sắp xếp dựa theo các lớp liên kết với nhau. Lớp đầu vào sẽ thu thập các mẫu đầu vào và lớp đầu ra sẽ thu nhận các phân loại hoặc tín hiệu đầu ra mà các mẫu đầu vào có thể phản ánh lại.

Mạng Neural Network là sự kết hợp của những tầng perceptron hay còn gọi là perceptron đa tầng. Và mỗi một mạng Neural Network thường bao gồm 3 kiểu tầng là:

* Tầng input layer (tầng vào): Tầng này nằm bên trái, thể hiện cho các đầu vào của mạng.
* Tầng output layer (tầng ra): Là tầng bênnó thể hiện cho những đầu ra của mạng.
* Tầng hidden layer (tầng ẩn): Tầng này nằm giữa tầng vào và tầng ra nó thể hiện cho quá trình suy luận logic của mạng.

# 2. Graph Neural Network.

Sự thành công gần đây của mạng nơ-ron đã thúc đẩy nghiên cứu về nhận dạng mẫu và khai thác dữ liệu. Các tác vụ học máy, như phát hiện đối tượng, dịch máy và nhận dạng giọng nói, đã được mang lại sức sống mới với các mô hình học sâu end-to-end như CNN hoặc RNN. Deep Learning rất tốt trong việc nắm bắt các mẫu ẩn của dữ liệu Euclid (hình ảnh, văn bản, video). Nhưng **còn các ứng dụng trong đó dữ liệu được tạo ra từ các miền không phải Euclide, được biểu diễn dưới dạng đồ thị với các mối quan hệ phức tạp và sự phụ thuộc lẫn nhau giữa các đối tượng?** **Đó là nơi xuất hiện của**[Graph Neural Networks](https://towardsdatascience.com/understanding-graph-convolutional-networks-for-node-classification-a2bfdb7aba7b)**(GNN)**. Chúng ta sẽ bắt đầu với lý thuyết đồ thị và các định nghĩa cơ bản.

**Một số dạng dữ liệu Graph trong tự nhiên**

Có thể các bạn đã quen thuộc với một số loại dữ liệu biểu đồ, chẳng hạn như mạng lưới quan hệ xã hội. Tuy nhiên, đồ thị là một đại diện dữ liệu cực kỳ hữu ích và tổng quát, bọn mình sẽ hiển thị hai loại dữ liệu mà bạn có thể không nghĩ rằng có thể được biểu diễn dưới dạng đồ thị: hình ảnh và văn bản.

**Ảnh dưới dạng graphs**

Chart, scatter chart

Description automatically generatedMột cách trực quan hóa khả năng kết nối của một đồ thị là thông qua ma trận kề của nó. Ta sắp xếp thứ tự các nút, trong trường hợp này là mỗi nút 25 pixel trong một hình ảnh 5x5 đơn giản của một khuôn mặt cười và điền vào ma trận nnodes × nnodes với đầu vào là những cặp nodes có chung cạnh với nhau.

**Văn bản dưới dạng graphs**

Chart

Description automatically generatedTa có thể số hóa văn bản bằng cách liên kết các chỉ số với từng ký tự, từ hoặc mã thông báo và biểu diễn văn bản dưới dạng một chuỗi các chỉ số này. Điều này tạo ra một đồ thị có hướng đơn giản, trong đó mỗi ký tự hoặc chỉ mục là một nút và được kết nối qua một cạnh với nút theo sau nó.

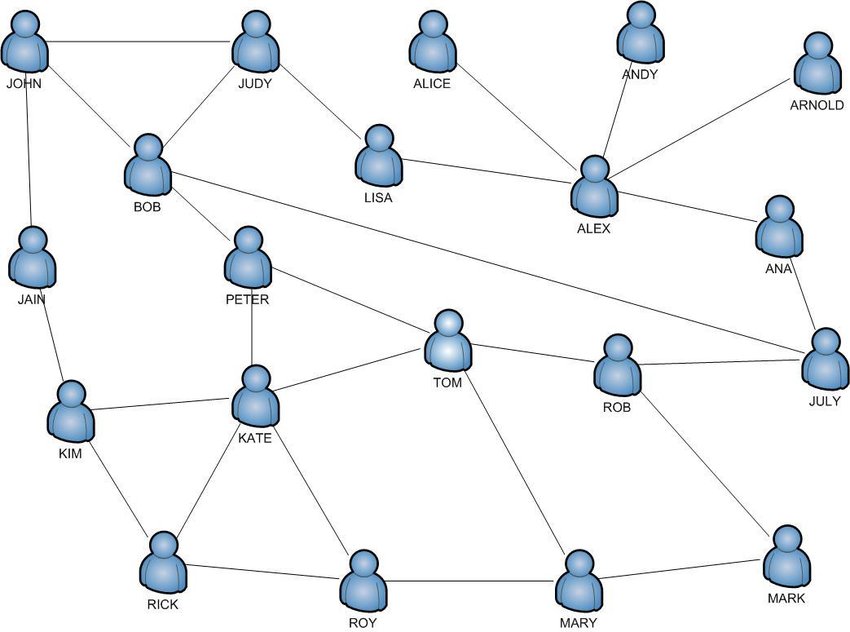
Tuy nhiên, trong thực tế, đây thường không phải là cách văn bản và hình ảnh được mã hóa: Việc biểu diễn những hình ảnh và văn bản dưới dạng Graph là dư thừa vì tất cả hình ảnh và tất cả văn bản sẽ có cấu trúc tương tự nhau. Ví dụ: hình ảnh có cấu trúc dải trong ma trận kề của chúng vì tất cả các nút (pixel) được kết nối trong một lưới. Ma trận kề cho văn bản chỉ là một đường chéo, bởi vì mỗi từ chỉ kết nối với từ trước và với từ tiếp theo. **(Lời giải thích của em Nghĩa ko cần đưa vào ppt)**

Đồ thị là một công cụ hữu ích để mô tả dữ liệu rất quen thuộc với chúng ta. Hãy chuyển sang dữ liệu có cấu trúc không đồng nhất hơn. Trong những ví dụ này, số lượng nút lân cận của mỗi nút là khác nhau (trái ngược với số lượng nút lân cận cố định của hình ảnh và văn bản). Dữ liệu này khó có thể biểu diễn dưới dạng khác ngoài biểu đồ **(Lời dẫn của em Nghĩa ko cần đưa vào ppt)**

**Chart, scatter chart

Description automatically generated**A picture containing text, crossword puzzle

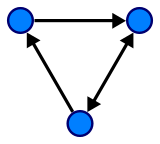
Description automatically generated**Phân tử dưới dạng Graphs**

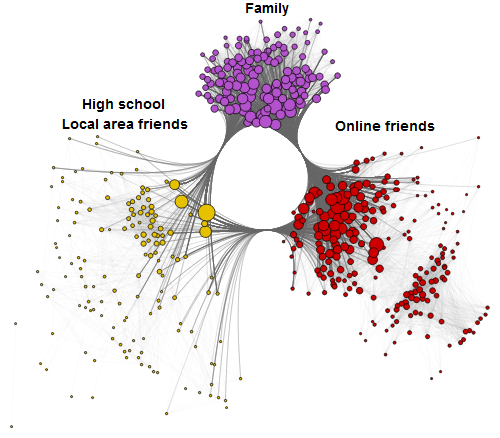
**Mạng lưới quan hệ xã hội dưới dạng Graphs**

Phần cơ bản nhất của GNN là Đồ thị.

Trong khoa học máy tính, **đồ thị là một cấu trúc dữ liệu bao gồm hai thành phần:**[các nút](https://towardsdatascience.com/understanding-graph-convolutional-networks-for-node-classification-a2bfdb7aba7b)**(đỉnh) và các cạnh** .

Một đồ thị **G** có thể được định nghĩa là **G = (V, E)** , trong đó V là tập hợp các nút và **E** là các cạnh giữa chúng.

Nếu có sự phụ thuộc định hướng giữa các nút thì các cạnh được định hướng. Nếu không, các cạnh sẽ bị vô hướng.

Biểu đồ có thể đại diện cho những thứ như mạng truyền thông xã hội hoặc các phân tử. Hãy coi các nút là người dùng và các cạnh là kết nối. Biểu đồ mạng xã hội có thể trông như hình ở trên.

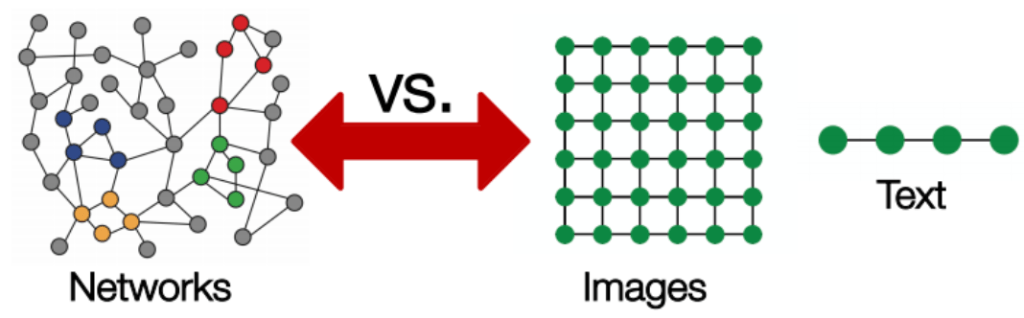
Một đồ thị thường được biểu diễn bằng **A**, một ma trận kề.

Nếu một đồ thị có n nút thì A có số chiều là **(n × n)** .

Đôi khi các nút có một tập hợp các tính năng (ví dụ, một hồ sơ người dùng). Nếu nút có **f** số đặc điểm, thì ma trận đặc điểm nút **X** có số chiều là **(n × f)** .

**Tại sao khó phân tích một biểu đồ?**

Lý do một biểu đồ khó phân tích là vì dữ liệu đồ thị rất phức tạp nên nó tạo ra rất nhiều thách thức cho các thuật toán học máy hiện có. Các công cụ Học máy và Học sâu thông thường chuyên về các kiểu dữ liệu đơn giản. Giống như các hình ảnh có cùng cấu trúc và kích thước, chúng ta có thể coi đó là biểu đồ lưới có kích thước cố định. Văn bản và lời nói là các chuỗi, vì vậy chúng ta có thể coi chúng là biểu đồ đường. Nhưng có nhiều đồ thị phức tạp hơn, không có dạng cố định, với kích thước thay đổi của các nút không có thứ tự, nơi các nút có thể có số lượng lân cận khác nhau. Nó cũng không giúp ích gì khi các thuật toán học máy hiện tại có một giả định cốt lõi rằng các phiên bản là độc lập với nhau. Điều này là sai đối với dữ liệu biểu đồ, vì mỗi nút có liên quan đến các nút khác bằng các liên kết thuộc nhiều loại khác nhau.



Graph Neural Network (GNN) là một lớp phương pháp học sâu được thiết kế để xử lý và thực hiện suy luận trên dữ liệu được biểu diễn dưới dạng đồ thị. GNN là mạng nơ-ron có thể được áp dụng trực tiếp vào biểu đồ và cung cấp một cách dễ dàng để thực hiện các tác vụ dự đoán cấp nút, cấp cạnh và cấp đồ thị. GNN có thể làm những gì mà Convolutional Neural Network (CNN) không làm được.

**Deep learning**

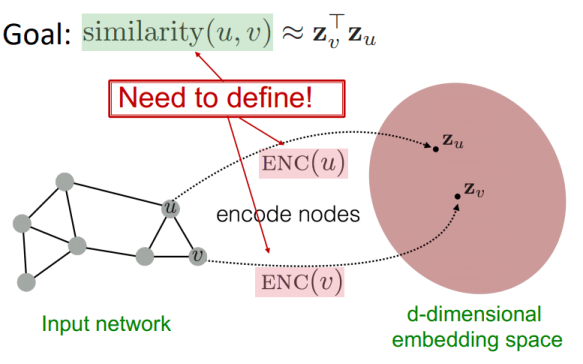
Trong [lý thuyết đồ thị](https://towardsdatascience.com/graph-deep-learning/home) , chúng ta thực hiện khái niệm Nhúng nút (Node Enbedding). Nó có nghĩa là ánh xạ các nút tới không gian nhúng một chiều (không gian có chiều thấp chứ không phải là kích thước thực của đồ thị), để các nút tương tự trong đồ thị được nhúng gần nhau.

Mục tiêu của chúng ta là ánh xạ các nút sao cho sự tương đồng trong không gian nhúng xấp xỉ mức độ tương tự trong mạng.

Hãy xác định u và v là hai nút trong đồ thị.

x u và x v là hai vectơ đặc trưng.

Bây giờ chúng ta sẽ xác định hàm mã hóa Enc (u) và Enc (v) , chuyển đổi các vectơ đặc trưng thành z u và z v .

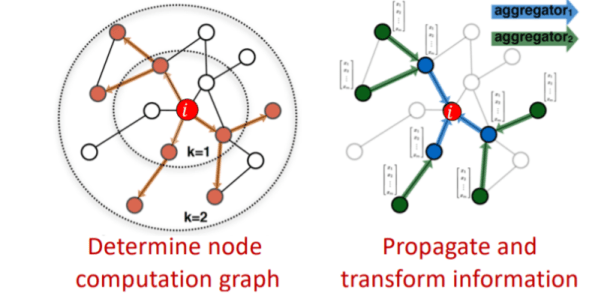


Chức năng bộ mã hóa sẽ có thể thực hiện:

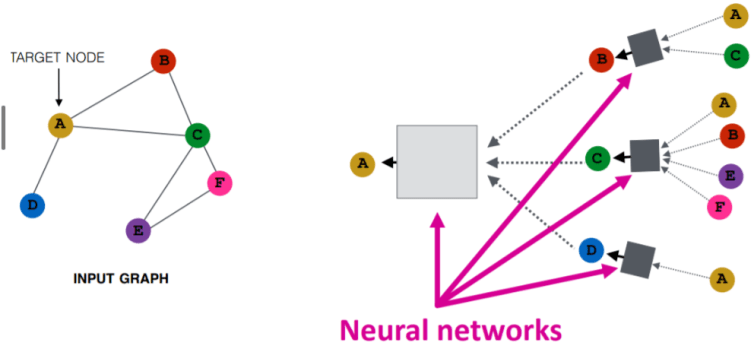
* Địa phương (vùng lân cận mạng cục bộ)
* Thông tin tổng hợp
* Xếp chồng nhiều lớp (tính toán)

Có thể lấy được thông tin lân cận bằng cách sử dụng đồ thị tính toán (Computational Graph). Như trong đồ thị bên dưới, ***i*** là node màu đỏ nơi chúng ta thấy cách node này được kết nối với các hàng xóm của nó và các hàng xóm của những người hàng xóm đó. Chúng ta sẽ thấy tất cả các liên kết có thể có và tạo thành một biểu đồ tính toán.

Bằng cách này, ta biết được cấu trúc và đồng thời lấy thông tin về thuộc tính.

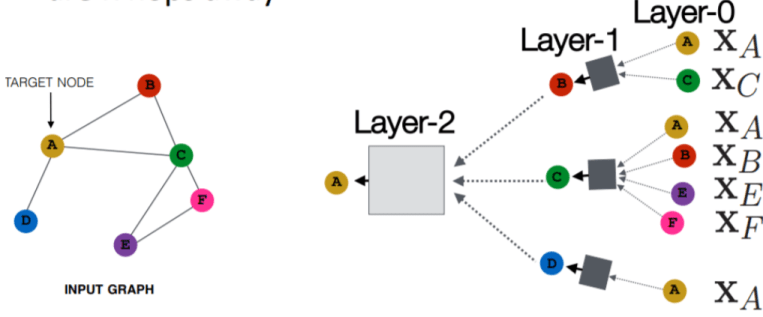


Sau khi thông tin liền kề lưu trữ biểu đồ tính toán, ta bắt đầu tổng hợp. Điều này về cơ bản được thực hiện bằng cách sử dụng Neural Networks.



Neural Networks được biểu diễn bởi những hộp màu xám. Chúng yêu cầu các tổng hợp phải bất biến theo thứ tự, như tổng, trung bình, tối đa, bởi vì chúng là các hàm hoán vị-bất biến. Thuộc tính này cho phép thực hiện tổng hợp.

Hãy chuyển sang **quy tắc lan truyền thuận** trong GNN. Nó xác định cách thông tin từ đầu vào sẽ đi đến phía đầu ra của Neural Networks.



Mỗi nút đều có một vector đặc trưng.

Ví dụ, ( ***X A*** ) là một vectơ đặc trưng của nút  ***A.***

Đầu vào là các vectơ đặc trưng đó và hộp xám sẽ lấy hai vectơ đặc trưng ( ***X A*** và ***X c*** ), tổng hợp chúng lại rồi chuyển sang lớp tiếp theo.

Lưu ý rằng, ví dụ: đầu vào tại nút ***C*** là các thuộc tính của nút ***C*** , nhưng biểu diễn của nút ***C*** trong lớp 1 sẽ là một biểu diễn ẩn và trong lớp 2 nó sẽ là một biểu diễn tiềm ẩn khác.

Vì vậy, để thực hiện truyền tiến trong đồ thị tính toán này, chúng ta cần 3 bước:

**1. Khởi tạo các đơn vị kích hoạt:**



**2. Mọi lớp trong mạng:**

Text

Description automatically generated

Chúng ta có thể nhận thấy rằng có hai phần cho phương trình này:

* Phần đầu tiên về cơ bản là tính trung bình tất cả các nút lân cận của nút ***v*** .

Shape

Description automatically generated with medium confidence

* Phần thứ hai là nhúng lớp trước của nút ***v*** được nhân với độ chệch ***B k*** , là một ma trận trọng số có thể huấn luyện được và về cơ bản nó là sự tự kích hoạt vòng lặp cho nút ***v*** .



* **σ** : kích hoạt phi tuyến tính được thực hiện trên hai phần.

**3. Phương trình cuối cùng (ở lớp cuối cùng):**



Đó là sự nhúng sau ***K*** lớp của tổng hợp vùng lân cận.

Bây giờ, để đào tạo mô hình, chúng ta cần xác định một hàm mất mát trên các lần nhúng.

Chúng tôi có thể cung cấp các nhúng vào bất kỳ hàm mất mát nào và chạy stochastic gradient descent để huấn luyện các trọng số.

Việc đào tạo có thể không được giám sát hoặc giám sát:

* **Huấn luyện không giám sát:**

Chỉ sử dụng cấu trúc đồ thị: các nút tương tự có cách nhúng tương tự. Hàm mất mát không được giám sát có thể là tổn thất dựa trên mức độ gần nút trong biểu đồ hoặc các bước đi ngẫu nhiên.

* **Huấn luyện có giám sát:**

Huấn luyện mô hình cho một tác vụ được giám sát như phân loại nút, nút bình thường hoặc bất thường.

Tóm lại, trong phần này, chúng ta đã mô tả ý tưởng cơ bản về việc tạo nhúng nút bằng cách tổng hợp thông tin vùng lân cận.

Tiếp theo, ta sẽ thảo luận về Graph Convolutional Networks (GCN).

# 3. Graph Convolutional Networks

Trong vài năm gần đây, một số bài báo đã nghiên cứu lại vấn đề tổng quát hóa mạng nơ-ron để làm việc trên các đồ thị có cấu trúc tùy ý ( [Bruna và cộng sự](http://arxiv.org/abs/1312.6203) , ICLR 2014; [Henaff và cộng sự](http://arxiv.org/abs/1506.05163) , 2015; [Duvenaud và cộng sự](http://papers.nips.cc/paper/5954-convolutional-networks-on-graphs-for-learning-molecular-fingerprints) , NIPS 2015; [Li và cộng sự](https://arxiv.org/abs/1511.05493) , ICLR 2016; [Defferrard và cộng sự](https://arxiv.org/abs/1606.09375) , NIPS 2016; [Kipf & Welling](http://arxiv.org/abs/1609.02907) , ICLR 2017), một số người trong số họ hiện đang đạt được kết quả rất hứa hẹn trong các miền trước đây bị chi phối bởi, ví dụ: phương pháp dựa trên hạt nhân, đồ thị- các kỹ thuật chính quy hóa dựa trên cơ sở và các kỹ thuật khác.

**Mạng lưới biểu đồ (GCN)**

**GCN**  là một loại  **mạng nơ-ron tích tụ**  có  **thể hoạt động trực tiếp trên đồ thị**  và tận dụng thông tin cấu trúc của chúng.

nó giải quyết vấn đề phân loại các nút (chẳng hạn như tài liệu) trong một đồ thị (chẳng hạn như mạng trích dẫn), nơi các nhãn chỉ có sẵn cho một tập con nhỏ các nút (học bán giám sát).

Như tên gọi "Convolutions" cho thấy, ý tưởng là từ Hình ảnh và sau đó được đưa sang Đồ thị. Tuy nhiên, khi Hình ảnh có cấu trúc cố định thì Đồ thị phức tạp hơn nhiều.

**Ý tưởng chung về GCN** : Đối với mỗi nút, chúng ta nhận được thông tin về đặc điểm từ tất cả các nút lân cận và tất nhiên, đối tượng địa lý của chính nó. Giả sử chúng ta sử dụng hàm average (). Chúng tôi sẽ làm tương tự cho tất cả các nút. Cuối cùng, chúng tôi cung cấp các giá trị trung bình này vào một mạng nơ-ron.

trong thực tế, chúng ta có thể sử dụng các hàm tổng hợp phức tạp hơn là hàm trung bình. Chúng ta cũng có thể chồng nhiều lớp chồng lên nhau để có GCN sâu hơn. Đầu ra của một lớp sẽ được coi là đầu vào cho lớp tiếp theo.

* GCN được sử dụng để học bán giám sát trên biểu đồ.
* GCN sử dụng cả các tính năng nút và cấu trúc để đào tạo
* Ý tưởng chính của GCN là lấy trung bình có trọng số của tất cả các đặc điểm nút của hàng xóm (bao gồm cả chính nó): Các nút ở mức độ thấp hơn có trọng số lớn hơn. Sau đó, chúng tôi chuyển các vectơ đặc trưng kết quả thông qua một mạng nơ-ron để đào tạo.
* Chúng ta có thể chồng nhiều lớp hơn để làm cho GCN sâu hơn. Xem xét các kết nối còn lại cho các GCN sâu. Thông thường, chúng tôi đi cấp GCN 2 hoặc 3 lớp.

# 4. Application for GNNs

Dữ liệu có cấu trúc đồ thị hiện diện ở khắp mọi nơi. Các vấn đề mà GNN giải quyết có thể được phân thành các loại sau:

* **Phân loại nút:** nhiệm vụ ở đây là xác định nhãn của các mẫu (được biểu diễn dưới dạng nút) bằng cách xem nhãn của các mẫu lân cận của chúng. Thông thường, các vấn đề thuộc loại này được huấn luyện theo cách bán giám sát, với chỉ một phần của biểu đồ được gắn nhãn.
* **Phân loại đồ thị:** nhiệm vụ ở đây là phân loại toàn bộ đồ thị thành các loại khác nhau. Nó giống như phân loại hình ảnh, nhưng mục tiêu thay đổi thành miền đồ thị. Các ứng dụng của phân loại đồ thị rất nhiều; Từ việc xác định một protein có phải là một enzym hay không trong tin sinh học, đến phân loại các tài liệu trong NLP, hoặc phân tích mạng xã hội.
* **Trực quan hóa đồ thị:** là một lĩnh vực của toán học và khoa học máy tính, là giao nhau của lý thuyết đồ thị hình học và trực quan hóa thông tin. Nó liên quan đến việc trình bày trực quan các biểu đồ để tiết lộ các cấu trúc và sự bất thường có thể có trong dữ liệu và giúp người dùng hiểu được các biểu đồ.
* **Dự đoán liên kết:** ở đây, thuật toán phải hiểu mối quan hệ giữa các thực thể trong đồ thị và nó cũng cố gắng dự đoán liệu có mối liên hệ giữa hai thực thể hay không. Điều quan trọng trong mạng xã hội chính là suy ra các tương tác xã hội hoặc đề xuất những người bạn có thể có cho người dùng. Nó cũng đã được sử dụng trong việc khuyến nghị các vấn đề của hệ thống và trong việc dự đoán các hoạt động phạm tội.
* **Phân cụm đồ thị:** đề cập đến việc phân cụm dữ liệu dưới dạng đồ thị. Có hai hình thức phân cụm khác nhau được thực hiện trên dữ liệu đồ thị. Phân cụm theo đỉnh tìm cách gom các nút của biểu đồ thành các nhóm có mật độ kết nối dày đặc dựa trên trọng số cạnh hoặc khoảng cách cạnh. Dạng thứ hai của phân cụm đồ thị coi các đồ thị là các đối tượng được phân nhóm và phân cụm các đối tượng này dựa trên sự tương đồng.

**GNN – Computer Vision**

Sử dụng CNN thông thường, máy móc có thể phân biệt và xác định các đối tượng trong hình ảnh và video. Mặc dù vẫn còn cần nhiều sự phát triển để máy móc có được trực giác hình ảnh của con người. Tuy nhiên, các kiến ​​trúc GNN có thể được áp dụng cho các bài toán phân loại ảnh.

Một trong những vấn đề này là tạo đồ thị cảnh, trong đó mô hình nhắm tới phân tích cú pháp một hình ảnh thành một đồ thị ngữ nghĩa bao gồm các đối tượng và các mối quan hệ ngữ nghĩa của chúng. Cho một hình ảnh, các mô hình tạo đồ thị cảnh phát hiện và nhận dạng các đối tượng và dự đoán các mối quan hệ ngữ nghĩa giữa các cặp đối tượng.

Tuy nhiên, số lượng các ứng dụng của GNN trong thị giác máy tính vẫn đang tăng lên. Nó bao gồm sự tương tác giữa con người với vật thể, phân loại hình ảnh ít lần chụp và hơn thế nữa.

**GNNs in Natural Language Processing**

Chúng ta biết rằng văn bản là một loại dữ liệu tuần tự có thể được mô tả bằng RNN hoặc LSTM. Tuy nhiên, đồ thị được sử dụng nhiều trong các tác vụ NLP khác nhau, do tính tự nhiên và dễ biểu diễn của chúng. Gần đây, đã có một sự gia tăng quan tâm đến việc áp dụng GNN cho một số lượng lớn các vấn đề NLP như phân loại văn bản, khai thác ngữ nghĩa trong dịch máy, định vị địa lý của người dùng, trích xuất quan hệ hoặc trả lời câu hỏi. Chúng ta biết rằng mọi nút là một thực thể và các cạnh mô tả mối quan hệ giữa chúng. Trong nghiên cứu NLP, vấn đề trả lời câu hỏi không phải là mới. Nhưng nó bị giới hạn bởi cơ sở dữ liệu hiện có. Dù vậy, với các kỹ thuật như GraphSage (Hamilton và cộng sự), các phương pháp có thể được tổng quát hóa cho các nút chưa từng thấy trước đó.

**GNNs in traffic**

Dự báo tốc độ, lưu lượng giao thông hoặc mật độ đường trong mạng lưới giao thông về cơ bản là quan trọng trong một hệ thống giao thông thông minh. Chúng tôi có thể giải quyết vấn đề dự đoán lưu lượng bằng cách sử dụng STGNN. Xem xét mạng lưới giao thông như một biểu đồ không gian-thời gian trong đó các nút là các cảm biến được lắp đặt trên đường, các cạnh được đo bằng khoảng cách giữa các cặp nút và mỗi nút có tốc độ giao thông trung bình trong một cửa sổ là các Input thuộc tính động.

**GNNs in chemistry**

Các nhà hóa học có thể sử dụng GNN để nghiên cứu cấu trúc đồ thị của phân tử hoặc hợp chất. Trong các đồ thị này, các nút là các nguyên tử, và các cạnh - các liên kết hóa học.