MÔ HÌNH MẠNG NƠRON GNN CHO XỬ LÝ DỮ LIỆU BIỂU DIỄN DƯỚI DẠNG ĐỒ THỊ

Lê Anh Tú[1*], Nguyễn Quang Hoan[2], Nguyễn Văn Nghiêm[1], Lê Sơn Thái[1]
[1]Trường ĐH Công nghệ Thông tin và Truyền thông – ĐH Thái Nguyên
[2]Học viện Công nghệ Bưu chính Viễn thông

TÓM TẮT

Bài báo này trình bày tổng quan về mô hình mạng nơron đồ thị (GNN) và khả năng ứng dụng của nó. Đây là một mô hình mạng nơron mới, phát triển từ mô hình mạng nơron đệ quy. GNN được thiết kế đặc biệt để xử lý dữ liệu biểu diễn dưới dạng đồ thị, một dạng dữ liệu phổ biến trong các lĩnh vực thị giác máy, hóa học phân tử, sinh học phân tử, nhận dạng mẫu và khai phá dữ liệu... Bài báo đề cập tới nguồn gốc của mạng GNN, kiến trúc mạng, thuật toán học và một số ứng dụng thực tiễn, đồng thời chỉ ra các đánh giá về kết quả, tính khả thi và định hướng nghiên cứu tiếp theo.

Từ khóa: *Mạng nơron nhân tạo, mạng nơron đồ thị, mạng đệ quy, mạng truy hồi, mạng truyền thẳng.*

GIỚI THIỆU

Để biểu diễn mối quan hệ tự nhiên của dữ liệu, người ta thường sử dụng cấu trúc dữ liệu đồ thị. Cấu trúc dữ liệu này được áp dụng nhiều trong các lĩnh vực thị giác máy, hóa học phân tử, sinh học phân tử, nhận dạng mẫu, phân tích ảnh, xử lý ngôn ngữ tự nhiên và khai phá dữ liệu... Ứng dụng thuộc các lĩnh vực này được chia làm hai loại: graph-focused và node-focused. Ví dụ, các ứng dụng phân lớp ảnh thuộc loại graph-focused, còn ứng dụng nhận dạng đối tượng trong ảnh thuộc loại node-focused. Tuy nhiên, cấu trúc đồ thị tương đối phức tạp và được chia làm nhiều loại như: đồ thị có hướng, vô hướng, có chu trình và không có chu trình… nên việc nghiên cứu các mô hình tính toán phù hợp là cần thiết. Trong lĩnh vực mạng nơron nhân tạo, đã có một vài mô hình mạng áp dụng cho dạng dữ liệu này. Ví dụ như, mạng nơron đệ quy (Recursive neural networks-RNN) [1][2] của M. Gori. Nhưng RNN không trực tiếp xử lý dữ liệu đồ thị, mà trước đó phải tiền xử lý để ánh xạ dữ liệu đồ thị thành vector số thực [3]. Điều này làm mất đi một số thông tin quan trọng như quan hệ về hình trạng giữa các nút trong đồ thị. Ngoài ra, RNN chỉ có thể xử lý dạng đồ thị có hướng và không có chu trình.

Năm 2005, M. Gori và đồng nghiệp tiếp tục đề xuất mô hình mạng nơron đồ thị (Graph Neural Network-GNN) [4][8] phát triển từ RNN, cho phép xử lý trực tiếp hầu hết các dạng đồ thị, mà không cần tiền xử lý dữ liệu về dạng vector. GNN là mô hình mạng nơron có giám sát. Mỗi nơron trong GNN tương ứng với một nút trong đồ thị, các nơron được kết nối với nhau theo cách kết nối của các nút trong đồ thị và cập nhật trạng thái, trao đổi thông tin cho đến khi mạng đạt trạng thái cân bằng. Đầu ra của GNN được tính toán cục bộ tại mỗi nút dựa trên trạng thái của nút đó. Tương tự như mô hình mạng nơron tế bào [5][6] và Hopfield [7], GNN sử dụng cơ chế khuếch tán thông tin để đảm bảo mạng đạt trạng thái cân bằng. GNN đã được ứng dụng trong nhiều bài toán như định vị đối tượng [9], xếp hạng trang web [10], trích rút nội dung câu [11], phân lớp ảnh theo cấu trúc [12]…

Trong bài báo này, phần 2 trình bày sơ qua về mạng nơron đệ quy – tiền thân của GNN, phần 3 trình bày mô hình mạng GNN, phần 4 trình bày thuật toán huấn luyện của mạng, phần 5 trình bày một số ứng dụng và cuối cùng là phần kết luận.

MẠNG NƠRON ĐỆ QUY [1]

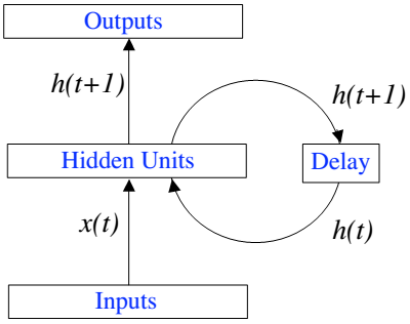
Mạng nơron đệ quy là một lớp các mạng nơron, dấu hiệu đặc trưng cơ bản của một mạng đệ quy (RNN) là mạng đó phải chứa ít

* *Tel: 0989 199088, Email: latu@ictu.edu.vn*

nhất một chu trình, khi được kích hoạt mạng sẽ có thể thực hiện việc lặp. Mạng có nhiều dạng kiến trúc khác nhau nhưng luôn có chung 2 đặc trưng quan trọng:

- Kết hợp một số mạng truyền thẳng đa lớp thành một hệ thống con.

- Tận dụng khả năng lập bản đồ phi tuyến của mạng truyền thẳng đa lớp cộng với một số dạng bộ nhớ.



Hình 1. *Kiến trúc tổng quát của RNN*

Hình 1 minh họa kiến trúc tổng quát của một mạng RNN với các lớp đầu vào, lớp đầu ra và lớp ẩn. Ngoài ra, "Delay" được hiểu là độ trễ về mặt tín hiệu cho bước hoạt động tiếp theo, kết quả tính toán hiện tại sẽ được sử dụng cho việc tính toán lần sau của mạng.

Như đã trình bày, RNN có thể xử lý một số bài toán với dạng dữ liệu đồ thị, nhưng không trực tiếp xử lý dữ liệu đồ thị, mà trước đó phải tiền xử lý để ánh xạ dữ liệu đồ thị thành dạng vector số thực [3]. Điều này làm mất đi một số thông tin quan trọng như quan hệ về hình trạng giữa các nút trong đồ thị chính vì vậy GNN đã được đề xuất để khắc phục những điểm yếu này.

MÔ HÌNH MẠNG NƠRON ĐỒ THỊ [8]

Đồ thị G là một cặp (N,E), trong đó N là tập các nút, E tập các cạnh của đồ thị. Gọi $ne[n]$ là tập các nút liền kề với nút n, $co[n]$ là tập các cạnh có nút là n. Các nút và cạnh của đồ thị có thể được gắn nhãn là các vector số thực, biểu diễn các đặc trưng hoặc các mối quan hệ giữa chúng. Nhãn của nút n ký hiệu là $l_n \in R^{l_N}$, nhãn của cạnh (n_1, n_2) ký hiệu là $l_{(n1,n2)} \in R^{l_E}$, với l_N tập các nhãn của nút, l_E tập các nhãn của cạnh. Các đồ thị được xét ở một trong hai dạng có chỉ số và không có chỉ số. Mỗi láng giềng u của nút n trong đồ thị có chỉ số được gán một định danh duy nhất $v_n(u)$ bằng hàm $v_n: ne[n] \to \{1,..,|N|\}$ để chỉ ra vị trí logic của nó.

Gọi H là tập các đồ thị, $\tilde{N}$ là tập con các nút của H, giả sử có tập học sau:

$H=\{(G_i, n_{i,j}, t_{i,j}) | G_i=(N_i, E_i) \in G; n_{i,j} \in N_i; t_{i,j} \in R^m, 1 \le i \le p, 1 \le j \le q_i\}$

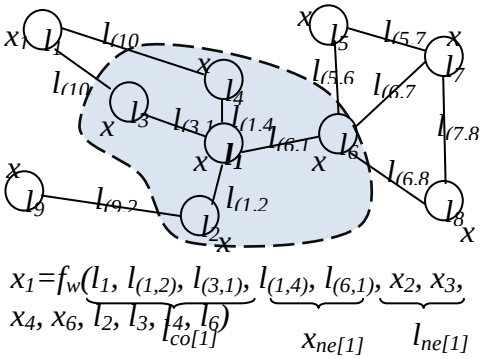
Trong đó, $n_{i,j} \in N_i$ là nút thứ j của tập nút N_i, $t_{i,j}$ là đầu ra mong muốn của nút $n_{i,j}$, $p \le |H|$, $q_i \le |N_i|$. Mạng được thiết kế theo nguyên tắc mỗi đỉnh trong đồ thị tương ứng với một nút (nơron) trong mạng. Một nút n trong mạng sẽ được gắn một giá trị trạng thái x_n, và có một giá trị đầu ra o_n được xác định bởi:

$$x_n = f_w(l_n, l_{co[n]}, x_{ne[n]}, l_{ne[n]})$$
$$o_n = g_w(x_n, l_n) \quad (1)$$

với f_w là hàm truyền, g_w là hàm đầu ra, l_n là nhãn nút n, $l_{co[n]}$ là nhãn của các cạnh có đỉnh là n, $x_{ne[n]}$ là trạng thái các nút láng giềng với n, $l_{ne[n]}$ là nhãn các nút láng giềng của n. Hình 2 minh họa cách xác định trạng thái x_1 của nút 1.

Trong trường hợp đồ thị không đánh chỉ số, hàm f_w của (1) được thay thế bởi:

$$x_n = \sum_{u \in ne[n]} h_w\left(l_n, l_{(n,u)}, x_u, l_u\right), n \in N \quad (2)$$



Hình 2. *Biểu diễn trạng thái x1 của nút 1 dựa vào thông tin láng giềng của nó*

Để thực hiện mô hình mạng GNN cần đảm bảo:

Có phương pháp thực hiện công thức (1)

Có thuật toán học phù hợp để điều chỉnh f_w và g_w sử dụng tập dữ liệu huấn luyện

Định lý điểm cố định của Banach [13] đã chứng minh tồn tại duy nhất (1) và gợi ý cách tính trạng thái kế tiếp của *x* theo (3):

$$x(t+1) = F_w(x(t),l) \quad (3)$$

Thực tế, công thức (3) sử dụng phương pháp lặp Jacobi giải phương trình phi tuyến [14] nên trạng thái và đầu ra kế tiếp được tính:

$$\begin{aligned} x_n(t+1) &= f_w(l_n, l_{co[n]}, x_{ne[n]}(t), l_{ne[n]}) \\ o_n(t) &= g_w(x_n(t), l_n), n \end{aligned} \quad (4)$$

Như vậy, từ đồ thị ban đầu ta cần xây dựng một mô hình tính toán thỏa mãn công thức (4) bằng cách thay thế mỗi đỉnh trong đồ thị bởi một đơn vị tính toán f_w. Mô hình này được gọi là mạng mã hóa (Hình 3b). Mỗi đơn vị *n* lưu trữ trạng thái hiện tại $x_n(t)$, khi kích hoạt sẽ tính trạng thái $x_n(t+1)$ dựa vào nhãn của nó và các thông tin từ các láng giềng, và đầu ra của nó được tính bởi g_w. Khi f_w và g_w được thực hiện bằng các mạng truyền thẳng thì mạng mã hóa trở thành mạng hồi quy. Mô hình tính toán của GNN được biểu diễn như Hình 3c, trong đó mỗi lớp biểu diễn một lần lặp tính toán, các nút trong một lớp được sao chép lại từ các đơn vị trong mạng mã hóa.

Thuật toán học của mạng [8]

Quá trình học của mạng GNN được thực hiện theo hướng ước lượng tham số *w* với giá trị φ_w xấp xỉ với dữ liệu trong tập huấn luyện. Với hàm lỗi là hàm bậc hai được xác định như sau:

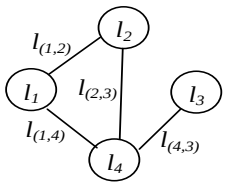
Thuật toán học dựa trên chiến lược giảm Gradient, gồm các bước sau:

Bước 1: Lặp lại việc cập nhật trạng thái $x_n(t)$ theo (4) đến lần lặp *T* thỏa mãn $x(T) \in x$, với *x* là vector tổng hợp của tất cả các trạng thái.

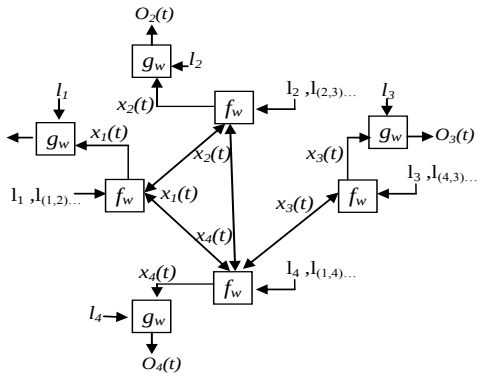
Bước 2: Tính gradient $\frac{\partial e_w(T)}{\partial w}$

Bước 3: Cập nhật trọng số *w* theo gradient tính ở bước 2.

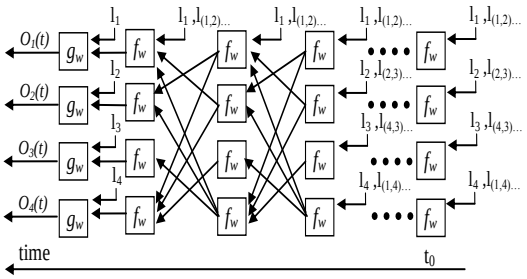
Theo đó thuật toán học có thể được minh họa như sau:



a) Đồ thị đầu vào



b) Mạng mã hóa



c) Mô hình tính toán của GNN

Hình 3. *Minh họa cách xây dựng một mạng GNN từ một đồ thị*

$$e_w = \sum_{i=1}^{p} \sum_{j=1}^{q_i} \left(t_{i,j} - \varphi_w \left(G_i, n_{i,j} \right) \right)^2 \quad (5)$$

Khởi tạo w;
x=Forward(w);
Repeat
$\frac{\partial e_w}{\partial w} = Backward(x, w)$;
$w = w - \lambda . \frac{\partial e_w}{\partial w}$;
x=Forward(w);
Until (điều kiện dừng)

Hàm *Forward(w)* như sau:

```
Khởi tạo x(0), t=0;
Repeat
    x(t+1)=Fw(x(t),l);
    t=t+1;
Until ||x(t)-x(t-1)|| ≤ εf
return x(t);
```

Hàm *Backward(x,w)* như sau:

```
o=Gw(x,lN);
A = ∂Fw/∂x (x,l);
b = ∂ew/∂o . ∂Gw/∂x (x,lN);
Khởi tạo z(0), t=0;
Repeat
    z(t)=z(t+1).A+b;
    t=t-1;
Until ||z(t-1)-z(t)||≤εb;
c = ∂ew/∂o . ∂Gw/∂w (x,lN);
d = z(t) . ∂Fw/∂w (x,l);
∂ew/∂w = c + d;
return ∂ew/∂w ;
```

Để tính g_w và f_w ta xét mạng trong hai trường hợp

GNN tuyến tính (không đánh chỉ số) [8]

Công thức (2) được tính như sau:

$h_w(l_n, l_{(n,u)}, x_u, l_u) = A_{n,u}x_u + b_n$ (6)

trong đó, $b_n \in R^s$, $A_{n,u} \in R^{s \times s}$ là đầu ra của hai mạng truyền thẳng tương ứng Forcing Network và Transition Network, các hàm $\phi_w : R^{2l_N + l_E} \to R^{s^2}$ và $\rho_w : R^{l_N} \to R^s$ được thực hiện bởi Forcing Network và Transition Network. Ta có:

$$A_{n,u} = \frac{\mu}{s \,|\, ne[u]\,|} . \Xi \quad (7)$$

$$b_n = \rho_w (l_n) \quad (8)$$

với $\mu \in (0,1)$, và $\Xi = resize\left(\phi_w \left(l_n, l_{(n,u)}, l_u\right)\right)$, *resize(.)* là toán tử chuyển đổi các thành phần của một vector *s* chiều thành ma trận *s×s*. Chú ý, trong trường hợp này $F_w(x,l)=Ax + b$, vì vậy $\frac{\partial F_w}{\partial x} = A$.

GNN phi tuyến (không đánh chỉ số) [8]

h_w được thực hiện bởi mạng truyền thẳng đa lớp, có thể xấp xỉ bất cứ hàm nào. Tuy nhiên, không phải tất cả các thông số đều được sử dụng vì phải đảm bảo hàm chuyển đổi tương ứng là một ánh xạ thu hẹp. Do đó, (5) được thêm vào một giá trị điều chỉnh như sau:

$$e_w = \sum_{i=1}^{p} \sum_{j=1}^{q_i} \left(t_{i,j} - \varphi_w \left(G_i, n_{i,j}\right)\right)^2 + \beta L\left(\| \frac{\partial F_w}{\partial x} \|\right)$$

trong đó, giá trị điều chỉnh $L(y)=(y-\mu)^2$ nếu $y>\mu$, ngược lại $L(y)=0$ và $\mu=(0,1)$.

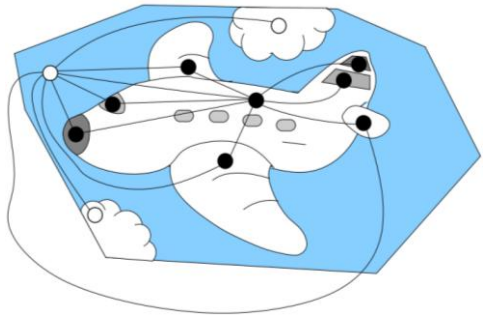
MỘT SỐ ỨNG DỤNG CỦA GNN

Xác định đối tượng trong ảnh [9]

Xác định đối tượng có trong ảnh hay không. Ảnh đưa vào được mã hóa về dạng đồ thị vùng liền kề (RAG). Các đỉnh là các vùng đồng nhất trong ảnh, sẽ có một cạnh nối 2 đỉnh nếu 2 vùng tương ứng là liền kề.

Nhãn nút bao gồm các đặc trưng của vùng ảnh như màu sắc (màu trung bình), hình học (diện tích, chu vi, trọng tâm)…trong khi đó

nhãn cạnh gồm đặc trưng màu sắc (sự khác biệt màu trung bình giữa 2 nút), hình học (khoảng cách giữa 2 trọng tâm của 2 nút, góc giữa 2 trục chính…)…

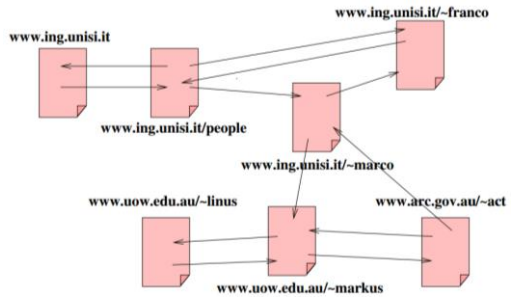


Hình 4. *Minh họa cách xây dựng đồ thị vùng liền kề từ ảnh đầu vào*

Nếu số chiều của trạng thái là 10, 10 lớp ẩn cho hàm f và 10 lớp ẩn cho hàm g, độ chính xác lên tới 90%

Xếp hạng các trang web [10]

Để có thể tiến hành xếp hạng các trang web với GNN người ta tiến hành mã hóa như sau:



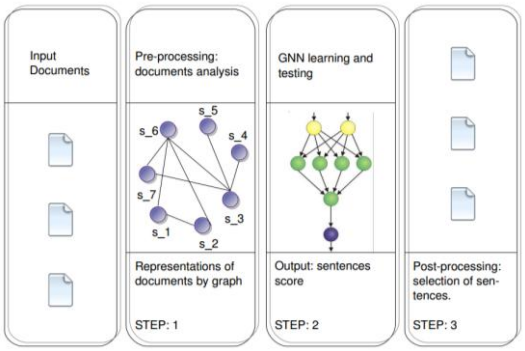
Hình 5. *Đồ thị hóa mối quan hệ giữa các trang web*

Một đồ thị với mỗi đỉnh tương ứng với một trang web và cạnh nối giữa 2 đỉnh sẽ được thiết lập nếu có liên kết giữa 2 trang web.

Sử dụng mạng truyền thẳng với 3 lớp (một lớp ẩn), 5 nơron ẩn với tập dữ liệu khoảng 4000 trang web thì việc xếp hạng theo chủ đề mất khoảng dưới 2 phút.

Trích rút câu [11]

Với mục đích khá thú vị là cho một tài liệu vào rút ra câu "quan trọng" để làm bản tóm tắt của tài liệu. Ứng dụng tiến hành mã hóa như sau:



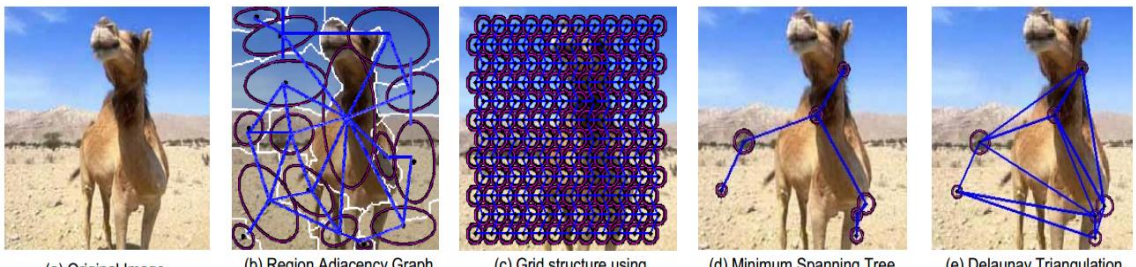
Hình 6. *Cách xây dựng đồ thị từ một tài liệu*

Mỗi tài liệu được tổ chức thành một đồ thị với mỗi đỉnh tương ứng với một câu, hai đỉnh sẽ được nối với nhau thành một cạnh nếu hai câu tương ứng có "độ tương tự" vượt ngưỡng cho trước.

Mặc dù chưa thể chính xác hoàn toàn nhưng kết quả đạt được khi thực hiện bằng GNN được đánh giá vượt trội so với các mô hình khác.

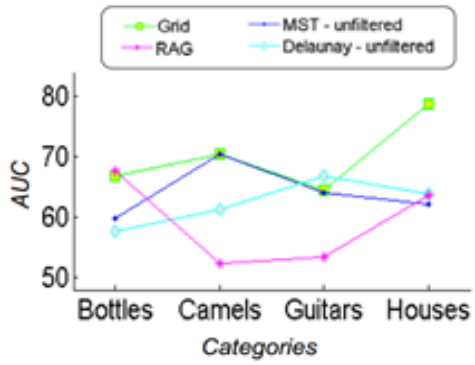
Phân loại ảnh theo cấu trúc [12]

Với mục đích phân loại ảnh theo cấu trúc vào danh mục cho trước. Ảnh đầu vào có thể được tổ chức dữ liệu như một số kiểu dưới đây:



Hình 7. *Một số cách mã hóa ảnh thành đồ thị*

Ứng dụng thực hiện với 350 ảnh cho độ chính xác dao động từ 60% đến 80%.



Ave AUC	RAG	Grid	MST Unfilt.	Delau Unfilt.
	59.35	70.21	64.16	62.43

KẾT LUẬN

GNN là một mô hình mạng nơron mới, mở rộng và kế thừa từ 2 mô hình mạng đệ quy và chuỗi Markov. Nó có thể xử lý được hầu hết các dạng dữ liệu đồ thị: đồ thị có thể có chu trình hoặc không, đồ thị có hướng, vô hướng hoặc hỗn hợp. Mô hình này dựa trên một cơ chế phục hồi và khuếch tán thông tin. Một thuật toán học cơ bản cũng đã được trình bày phần nào cho chúng ta thấy được độ phức tạp của mô hình. Bài báo cũng đã đề cập tới một số ứng dụng phổ biến có thể giải quyết bằng GNN. GNN là một mô hình mới cho vấn đề xử lý dữ liệu biểu diễn dưới dạng đồ thị nhưng lại có khả năng làm việc được với hầu hết các dạng đồ thị. Bằng cách tiếp cận mới mẻ cho những vấn đề tưởng như cũ kỹ (phân loại ảnh, định vị đối tượng trong ảnh hay xếp hạng các trang web…) nó hứa hẹn sẽ ngày càng được ứng dụng rộng rãi.

Tuy nhiên bản thân đồ thị là một dạng dữ liệu phức tạp, việc tiền xử lý để biểu diễn dữ liệu khác về dạng dữ liệu đồ thị lại càng phức tạp hơn. Ngoài ra, khối lượng tính toán theo mô hình GNN là khá lớn và phức tạp, gây ra nhiều khó khăn khi triển khai. Đây là những vấn đề cần cân nhắc và hứa hẹn sẽ mở ra nhiều hướng nghiên cứu trong thời gian tới.

TÀI LIỆU THAM KHẢO

1. P. Frasconi, M. Gori, and A. Sperduti, (September 1998) *"A General Framework for Adaptive Processing of Data Structures"*, IEEE Transactions on Neural Networks, vol. 9, no. 5, pp. 768-786.

2. A. Sperduti and A. Starita, (1997) *"Supervised Neural Networks for the Classification of Structures"*, IEEE Transactions on Neural Networks, vol. 8, pp. 429-459.

3. M. Bianchini, P. Mazzoni, L. Sarti, and F. Scarselli, (Sep. 2003) *"Face Spotting in Color Images Using Recursive Neural Networks"*, in Proc. 1st Int. Work-shop Artif. Neural Netw. Pattern Recognit., Florence, Italy, pp. 76–81.

4. Marco Gori, Gabriele Monfardini, Franco Scarselli, (2005) *"A New Model for Learning in Graph Domains"*, Proceedings of International Joint Conference on Neural Networks, Montreal, Canada, July 31 - August 4.

5. L. Chua and L. Yang, (Oct. 1988) *"Cellular Neural Networks: Theory"*, IEEE Trans. Circuits Syst., vol. CAS-35, no. 10, pp. 1257–1272.

6. L. Chua and L. Yang, *"Cellular Neural Networks: Applications"*, IEEE Trans. Circuits Syst., vol. CAS-35, no. 10, pp. 1273–1290, Oct. 1988.

7. J. Hopfield, (1982) "Neural Networks and Physical Systems with Emergent Collective Computational Abilities", Proc. Nat. Acad. Sci., vol. 79, pp.2554–2558.

8. Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, Gabriele Monfardini, (January 2009) *"The Graph Neural Network Model"*, IEEE Transactions on Neural Networks, vol. 20, no. 1,.

9. Gabriele Monfardini, Vincenzo Di Massa, Franco Scarselli, Marco Gori, *"Graph Neural Networks for Object Localization"*, ECAI 2006 G. Brewka et al. (Eds.), IOS Press, 2006.

10. Franco Scarselli, Sweah Liang Yong, Marco Gori, (2005) *"Graph Neural Networks for Ranking Web Pages"*, Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05).

11. Donatella Muratore, (2009) "Sentence Extraction by Graph Neural Networks", Thesis.

12. Quek, Zhiyong Wang, J. Zhang, Dagan Feng, (2011) *"Structural Image Classification with Graph Neural Networks"*, International Conference on Digital Image Computing Techniques and Applications (DICTA), pp. 416-421.

13. M. A. Khamsi, (2001) "An Introduction to Metric Spaces and Fixed Point Theory", New York: Wiley.

14. F. Scarselli and A. C. Tsoi, *"Universal Approximation Using Feedforward Neural Networks: A Survey of Some Existing Methods, and Some New Results"*, Neural Netw., vol. 11, no. 1, pp. 15–37, 1998.

SUMMARY

GNN NEURAL NETWORK MODEL TO PROCESS THE DATA IS PRESENT AS GRAPHICS

Le Anh Tu[1*], Nguyen Quang Hoan[2], Nguyen Van Nghiem[1], Le Son Thai[1]
[1]College of Information and Communication Technology – TNU
[2]Posts and Telecommunications Institute of Technology

This paper presents an overview of the graph neural network model and the possibility of its application. This is a new neural network model, developed from the recursive neural network model. GNN is specifically designed to process data which is represented in graphs, a common data format in the field of machine vision, molecular chemistry, molecular biology, pattern recognition and data mining... The paper not only refers to the origin of GNN networks, network architectures, algorithm and a number of practical applications, but also indicates the evaluation of the results, the feasibility of further research.

Keywords: *Artificial neural network, graph neural network, recursive neural network, recurrent neural network, feedforward neural network.*

Ngày nhận bài:25/01/2014; Ngày phản biện:10/02/2014; Ngày duyệt đăng: 26/02/2014
Phản biện khoa học: *TS. Vũ Đức Thái – Trường ĐH Công nghệ Thông tin & Truyền thông - ĐHTN*

[*] *Tel: 0989 199088, Email: latu@ictu.edu.vn*