

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA ĐIỆN – ĐIỆN TỬ
BỘ MÔN VIỄN THÔNG

-----o0o-----



ĐỒ ÁN MÔN HỌC
XÂY DỰNG MẠNG NEURAL TÍCH CHẬP DÙNG PHƯƠNG
PHÁP CẤU TRÚC ĐỂ PHÂN TÍCH ẢNH Y KHOA

GVHD: GS. TS. Lê Tiến Thường

Sinh viên: Tô Thanh Nhã

MSSV: 1810380

TP. Hồ Chí Minh, tháng 8, năm 2021

MỤC LỤC

DANH SÁCH HÌNH MINH HỌA	ii
1. GIỚI THIỆU	3
1.1 Đặt vấn đề	3
1.2 Mục đích nghiên cứu.....	4
1.3 Tổng quan tài liệu.....	4
1.4 Đối tượng và mục tiêu nghiên cứu	6
2. CƠ SỞ LÝ THUYẾT.....	7
2.1 Lý thuyết cấu trúc.....	7
2.2 Biến đổi Fourier và bộ lọc cấu trúc	8
2.3 Mạng neural tích chập	10
2.4 Graph CNN	11
3. NỘI DUNG VÀ PHƯƠNG PHÁP NGHIÊN CỨU	15
3.1 Chuẩn bị tập dữ liệu	15
3.2 Xây dựng cấu trúc ứng dụng khảo sát ảnh y khoa	16
3.3 Xác định hàm truyền thuận	17
3.4 Xây dựng hàm mất mát và lan truyền ngược	18
3.5 Tiến hành huấn luyện và đánh giá kết quả	20
4. KẾT QUẢ THỰC HIỆN.....	21
5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	24
5.1 Kết luận	24
5.2 Hướng phát triển	24
6. TÀI LIỆU THAM KHẢO.....	26

DANH SÁCH HÌNH MINH HỌA

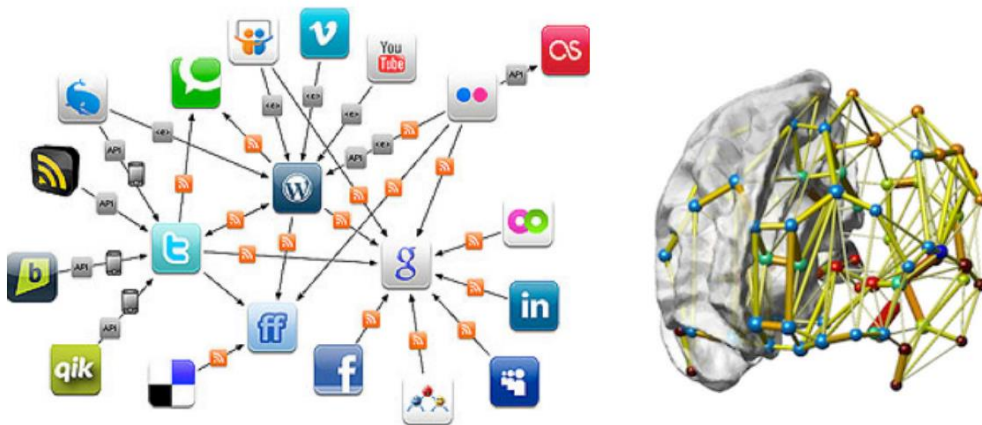
Hình 1-1: GSP trong mạng xã hội và y sinh	3
Hình 2-1: Cấu trúc tín hiệu thời gian có $N = 24$	7
Hình 2-2: Cấu trúc tín hiệu hình ảnh $N = 9$	7
Hình 2-3: Toán tử dịch cấu trúc với một cấu trúc $N = 6$	8
Hình 2-4: Mô hình mạng neural	10
Hình 2-5: Lớp kết hợp kích thước 2×2	11
Hình 2-6: Kiến trúc Graph CNN	12
Hình 2-7: Mô hình GCNN với 2 lớp	12
Hình 2-8: Tổng hợp cấu trúc phân biệt	13
Hình 2-9: Tổng hợp cấu trúc sắp xếp.....	13
Hình 3-1: Ảnh chụp chứa bất thường trong não.....	15
Hình 3-2: Ảnh chụp não không có bất thường	16
Hình 4-1: Quá trình tối ưu sai số.....	21
Hình 4-2: Kết quả ảnh không có bất thường	21
Hình 4-3: Kết quả ngõ ra sau khi huấn luyện.....	22
Hình 4-4: Các kết quả bị quá khớp.....	23

1. GIỚI THIỆU

1.1 Đặt vấn đề

- Ngày nay, chúng ta đang sống trong thế giới mà dữ liệu có mặt ở khắp mọi nơi, mọi lúc và đa dạng về hình thức. Không chỉ các tín hiệu truyền thống như giọng nói, âm thanh, hình ảnh, video, y sinh, radar, ... các tín hiệu còn được tạo ra từ các ứng dụng IoT, hàng tỷ người dùng điện thoại di động, hệ thống giao thông, bệnh viện, các mạng xã hội, ... Các dữ liệu lớn (Big Data) thường được đặc trưng bởi các đặc điểm cụ thể dù được tạo ra từ các nguồn dữ liệu phân tán, có thể mô tả bằng một cấu trúc nào đó [1].
- Xử lý tín hiệu cấu trúc (**Graph Signal Processing - GSP**) là lĩnh vực có sự phát triển đáng kể, nổi bật trong những năm gần đây và là một xu hướng chắc chắn tiếp tục phát triển mạnh mẽ trong những năm tới, liên quan đến các hệ thống phức tạp, không gian phi Euclid (non-Euclidean), dữ liệu lớn được mô tả bởi các phần tử có liên kết với nhau. Các ứng dụng giúp ích con người trong nhiều lĩnh vực khoa học, thông tin, truyền thông, y sinh, giao thông, kinh tế, mạng xã hội [2].

Online social media



Hình 1-1: GSP trong mạng xã hội và y sinh

- Với mục tiêu phát hiện các nguyên tắc giữa các đối tượng khác nhau từ các dữ liệu phân tán, học tập qua mạng lưới thông tin, từ đó mô hình hóa, nhận dạng đối tượng, ra quyết định tối ưu [3].

1.2 Mục đích nghiên cứu

- Hiện nay, trình độ khoa học kỹ thuật ngày càng phát triển, kéo theo sự nâng cao chất lượng của các dịch vụ y tế. Các thiết bị kiểm tra sức khỏe có thể nhanh chóng cho ra dữ liệu về người bệnh dưới dạng đồ thị điện não, điện tim, ảnh X-ray, MRI, CT, PET, ... Nhưng để đưa ra kết luận chính xác từ những thông tin trên cần có chuyên môn từ bác sĩ, thông qua đọc dữ liệu và phân tích, chuẩn đoán. Công việc này phải thực hiện tuần tự và mất nhiều thời gian, công sức, phụ thuộc vào kinh nghiệm, chuyên môn chủ quan của bác sĩ.
- Đề tài được thực hiện nhằm mục đích tạo ra mô hình mạng học tập từ cấu trúc graph có khả năng phát hiện tự động, chỉ ra các bất thường trên hình ảnh y khoa, hỗ trợ cho các bác sĩ trong việc sàng lọc các dữ liệu hình ảnh, sớm cảnh báo các nguy hiểm từ các điểm bất thường của bệnh nhân.

1.3 Tổng quan tài liệu

- Các phương pháp được sử dụng phổ biến hiện nay cho các ảnh y sinh là mạng neural tích chập (CNN) [4] hay phân tích thành phần đặc chính (PCA), phân cụm K – means [5]. Phương pháp CNN tận dụng ưu thế về phép chập trong miền không gian để khai thác các đặc điểm liên kết giữa các điểm ảnh lân cận với nhau, và các thông số mô hình sẽ được học qua nhiều lớp mạng, tuy nhiên hình ảnh y sinh đòi hỏi dung lượng lớn, khó đào tạo với lớp mạng dày đặc, hơn nữa các đặc điểm về y sinh không chỉ có có liên kết lân cận mà còn liên hệ với các điểm ảnh ở xa theo nhiều dạng cấu trúc hình học đặc biệt như xương, mô tế bào. Trong khi đó, PCA giúp giảm số chiều của ảnh qua phép chiếu lên các không gian trực giao, để vẫn giữ những đặc trưng chấp nhận được của dữ liệu và K – means phân nhóm dữ liệu ra các vùng giúp giảm đáng kể kích thước. Nhưng cả hai phương pháp phù hợp hơn với bài toán phân loại và nhận dạng đối tượng, không hoạt động tốt khi phát hiện các bất thường trên ảnh, đặc biệt là các điểm cục bộ tương đối nhỏ.
- Nhiều phương pháp khác ra đời tận dụng lợi thế về giảm số chiều của PCA và đặc điểm tích chập để khai thác các tính chất của điểm ảnh lân cận, nhưng mở rộng hơn cho các điểm ảnh lân cận là một đồ hình (graph) hay cấu trúc biểu đạt mối liên kết giữa các điểm ảnh (không nhất thiết lân cận) và mang những đặc trưng chính tín hiệu, khi đó các phân tử không được đối xử đồng đều (Euclidean)

mà được xử lý không bình thường hay xử lý không gian non-Euclidean [6]. Trong tài liệu tham khảo [6] là phương pháp CNN trên miền cấu trúc (GCNN) là một ví dụ tương đối đơn giản với ảnh cố tình tạo sự không đồng đều bằng cách bỏ đi ngẫu nhiên các điểm ảnh và kết quả sử dụng GCNN mang lại hiệu quả đáng kể. Thuật toán CNN hoạt động tốt với hình ảnh, có cấu trúc đồng nhất, nhưng nhiều ứng dụng với cấu trúc không đồng đều và đặc trưng bởi một đồ thị hoặc đa tạp để mô hình như mạng xã hội, lưới giao thông, cấu trúc sinh học, ... CNN hoạt động không cần chú trọng đến ý nghĩa về cấu trúc của đối tượng, do đó bỏ lỡ nhiều đặc tính quan trọng và tính toán công kênh do có quá nhiều sự liên kết là không cần thiết, hoặc không có nhiều ảnh hưởng. Trong khi đó Graph CNN (GCNN) quan tâm đến các mối liên hệ của các đối tượng trong tập dữ liệu, do đó có thể giúp giảm số chiều cần tính toán mà vẫn giữ được các đặc trưng cơ bản. Đồ hình tạo thành tương tự việc tạo sóng mang cho tín hiệu dữ liệu, nắm giữ kiến trúc cơ bản về mối quan hệ không gian giữa các đỉnh và cho phép nhiều các toán tử xử lý tín hiệu được thực hiện trên dữ liệu qua đồ hình, chẳng hạn như lọc wavelet, tích chập và biến đổi Fourier. Kết hợp các kỹ thuật xử lý tín hiệu đồ thị này với học sâu, có thể tìm hiểu trong các miền có khoảng cách không đều nhau (irregular). Kết quả [6] bước đầu đưa ra khái niệm nên chỉ thực hiện trên dữ liệu giả ngẫu nhiên, chưa thực sự đưa ra một ứng dụng cụ thể nào, các tài liệu [7] và [8] lần lượt ứng dụng GCNN trên ảnh y sinh cho phân đoạn tế bào và phân loại chứng viêm khớp gối chứng minh sự tiềm năng của mô hình này so với các phương pháp học tập truyền thống. Phương pháp mô tả cấu trúc trong [7] sử dụng khoảng cách từ điểm đang xét đến các điểm lân cận suy giảm theo hàm mũ có thể giảm quyết định được cho đối tượng cục bộ như các tế bào nhưng các bất thường trên phạm vi lớn hơn lại không hiệu quả vì vượt quá định nghĩa của cấu trúc. Các tác giả trong tài liệu [8], tạo ra mô hình với cấu trúc loại bỏ các ảnh hưởng về tuyến tính và xây dựng cấu trúc phi tọa độ, đi sâu vào bản chất thông qua học bán giám sát, tuy nhiên lại chưa có bước kết hợp để hạn chế số chiều và độ lớn dữ liệu – một ưu thế khi dùng cấu trúc, vẫn chưa khai thác triệt để.

1.4 Đối tượng và mục tiêu nghiên cứu

- Mục tiêu của đề tài từ bước đầu thu thập các dữ liệu về hình ảnh y khoa bao gồm cả bất thường và bình thường để tìm ra mô hình cấu trúc thể hiện mối liên hệ tương quan trong các bức ảnh. Từ mô hình graph đã có sẽ thông qua mạng neural tích chập đồ thị để huấn luyện một mạng có khả năng đáp ứng các mục đích đã đề ra.
- Đối tượng thực hiện là các ảnh y sinh như X quang, MRI, CT.
- Phạm vi nghiên cứu của đề tài chỉ tập trung trên các ảnh chấn thương (hay bất thường) trên xương, mô và cơ quan của con người thể hiện qua các ảnh y khoa.

2. CƠ SỞ LÝ THUYẾT

2.1 Lý thuyết cấu trúc

- Xét cấu trúc (graph): $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ với các thông số trong cấu trúc lần lượt là tập hợp các đỉnh, các cạnh và trọng số các liên kết.

$$\mathcal{V} = \{1, 2, \dots, N\} \quad (2-1)$$

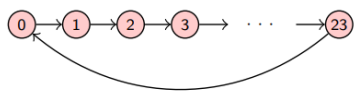
là tập hợp các đỉnh (vertices) hay các nút (nodes) của cấu trúc. $N = |\mathcal{V}|$ là số đỉnh khảo sát trong cấu trúc \mathcal{G} .

$$\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V} \quad (2-2)$$

là tập hợp các cạnh (edges) hay các liên kết giữa các đỉnh của cấu trúc \mathcal{G} .

$$\mathcal{W}: \mathcal{E} \rightarrow \mathbb{R}_+ \quad (2-3)$$

là các trọng số (weights) tương ứng w_{ij} với mỗi cạnh $(i, j) \in \mathcal{E}$. Trong trường hợp tổng quát trọng số liên kết không nhất thiết phải là số thực dương.



► **Unweighted** and **directed** graphs (e.g., time)

- $\mathcal{V} = \{0, 1, \dots, 23\}$
- $\mathcal{E} = \{(0, 1), (1, 2), \dots, (22, 23), (23, 0)\}$
- $\mathcal{W} : (n, m) \mapsto 1, \text{ for all } (n, m) \in \mathcal{E}$

Hình 2-1: Cấu trúc tín hiệu thời gian có $N = 24$

- Một số trường hợp đặc biệt:

+ Cấu trúc không trọng số (unweighted graph):

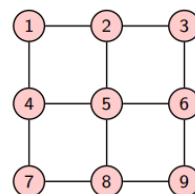
$$w_{ij} \in \{0, 1\}, \quad \forall (i, j) \in \mathcal{E} \quad (2-4)$$

+ Cấu trúc vô hướng (undirected graph):

$$w_{ij} = w_{ji}, \quad \forall (i, j) \in \mathcal{E} \quad (2-5)$$

· **Unweighted** and **undirected** graphs (e.g., image)

- $\mathcal{V} = \{1, 2, 3, \dots, 9\}$
- $\mathcal{E} = \{(1, 2), (2, 3), \dots, (8, 9), (1, 4), \dots, (6, 9)\}$
- $\mathcal{W} : (n, m) \mapsto 1, \text{ for all } (n, m) \in \mathcal{E}$



Hình 2-2: Cấu trúc tín hiệu hình ảnh $N = 9$

Dễ dàng nhận thấy rằng, tín hiệu thời gian và hình ảnh là các trường hợp đặc biệt của cấu trúc với các đỉnh là các vị trí lấy mẫu và liên kết giữa các nút với nhau là đồng đều. Tín hiệu thời gian có cấu trúc dạng vòng (ring) trong khi hình ảnh có cấu trúc dạng lưới (grid).

- Ma trận kề (adjacency matrix) $\mathbf{A} \in \mathbb{R}^{N \times N}$ của cấu trúc đặc trưng cho sự liên kết giữa các nút trong cấu trúc đó:

$$\mathbf{A} = [A_{ij}] = \begin{cases} w_{ij}, & (i, j) \in \mathcal{E} \\ 0, & (i, j) \notin \mathcal{E} \end{cases} \quad (2-6)$$

- lân cận (in-neighbor) của nút i là tập hợp các nút có liên kết với nó:

$$\mathcal{N}_i = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\} \quad (2-7)$$

- Ma trận bậc (degree matrix) $\mathbf{D} \in \mathbb{R}^{N \times N}$ của cấu trúc là ma trận chéo, với mỗi phần tử trên đường chéo thành lập từ vector tổng các liên kết tại mỗi nút.

$$\mathbf{D} = \text{diag}(\mathbf{A}\mathbf{1}) \quad (2-8)$$

trong đó, $\mathbf{1} = [1 \ 1 \ \dots \ 1]^T$.

- Ma trận Laplacian (Laplacian matrix) $\mathbf{L} \in \mathbb{R}^{N \times N}$ được định nghĩa là:

$$\mathbf{L} = \mathbf{D} - \mathbf{A} \quad (2-9)$$

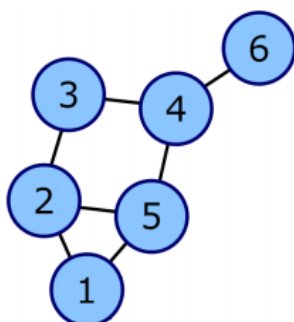
Ma trận Laplacian chuẩn hóa (Normalized Laplacian):

$$\mathbf{L}_n = \mathbf{D}^{-1/2} \mathbf{L} \mathbf{D}^{-1/2} \quad (2-10)$$

2.2 Biến đổi Fourier và bộ lọc cấu trúc

- Toán tử dịch cấu trúc (graph-shift operator): $\mathbf{S} \in \mathbb{R}^{N \times N}$ thể hiện cấu trúc cục bộ trong cấu trúc

$$S_{ij} = 0 \leftrightarrow \begin{cases} i \neq j \\ (i, j) \notin \mathcal{E} \end{cases} \quad (2-11)$$



$$\mathbf{S} = \begin{pmatrix} S_{11} & S_{12} & 0 & 0 & S_{15} & 0 \\ S_{21} & S_{22} & S_{23} & 0 & S_{25} & 0 \\ 0 & S_{23} & S_{33} & S_{34} & 0 & 0 \\ 0 & 0 & S_{43} & S_{44} & S_{45} & S_{46} \\ S_{51} & S_{52} & 0 & S_{54} & S_{55} & 0 \\ 0 & 0 & 0 & S_{64} & 0 & S_{66} \end{pmatrix}$$

Hình 2-3: Toán tử dịch cấu trúc với một cấu trúc $N = 6$

thông thường \mathbf{S} có thể được chọn là ma trận \mathbf{A} , ma trận \mathbf{L} , ...

- Ma trận S có thể chéo hóa (diagonalizable) sẽ có thể được phân tách thành:

$$\mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1} \quad (2-12)$$

trong đó, $\mathbf{\Lambda} = \text{diag}(\boldsymbol{\lambda}) = \text{diag}([\lambda_1, \lambda_2, \dots, \lambda_N]^T)$ và $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_N]$

Với λ_i và \mathbf{v}_i là các trị riêng (eigenvalue) và vector riêng (eigenvector) của ma trận \mathbf{S} , thể hiện các tần số cơ bản của cấu trúc.

- Tổng biến thiên (Total Variation): của một vector là đại lượng vô hướng không âm đặc trưng cho độ dao động của tín hiệu đó trên cấu trúc, được dùng để xếp thứ tự các tần số trong cấu trúc. Do các trị riêng đa phần đều là các giá trị phức nên để xác định một tần số là thấp hay cao dựa tổng biến thiên của vector riêng tương ứng (trị riêng có tổng biến thiên càng lớn thì đại diện cho tần số càng cao). Có nhiều định nghĩa khác nhau về tổng biến thiên của một vector, như dạng toàn phương của ma trận Laplacian. Một định nghĩa khác về tổng biến thiên được dựa trên ma trận kề \mathbf{A} :

$$TV(\mathbf{v}) = \|\mathbf{v} - \mathbf{A}^{norm}\mathbf{v}\|_1 \quad (2-13)$$

trong đó, chuẩn hóa ma trận $\mathbf{A}^{norm} = \frac{1}{\lambda_{max}} \mathbf{A}$

- Biến đổi Fourier cấu trúc (Graph Fourier Transform - GFT) của tín hiệu \mathbf{x} được định nghĩa là:

$$\tilde{\mathbf{x}} = \mathbf{V}^{-1}\mathbf{x} \quad (2-14)$$

- Biến đổi ngược Fourier cấu trúc (Inverse Graph Fourier Transform - iGFT) là:

$$\mathbf{x} = \mathbf{V}\tilde{\mathbf{x}} \quad (2-15)$$

Nếu \mathbf{S} trực giao, ta có: $\mathbf{V}^{-1} = \mathbf{V}^H \Rightarrow \mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^H$ và $\tilde{\mathbf{x}}_k = \mathbf{v}_k^H \mathbf{x}$.

Trong nhiều ứng dụng thường xem xét ma trận đối xứng nên có được $\mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$.

- Định lý Parseval:

$$\|\mathbf{x}\|^2 = \|\tilde{\mathbf{x}}\|^2 \quad (2-16)$$

- Bộ lọc cấu trúc (Graph Filter) bậc L với các hệ số $\mathbf{h} = [h_0, h_1, \dots, h_L]^T$ có ngõ vào \mathbf{x} và ngõ ra \mathbf{y} là:

$$\begin{aligned} \mathbf{y} &= \mathbf{H}\mathbf{x} \\ \mathbf{H} &= \sum_{l=0}^L h_l \mathbf{S}^l = h_0 \mathbf{I} + h_1 \mathbf{S} + \dots + h_L \mathbf{S}^L \end{aligned} \quad (2-17)$$

- Đa thức bộ lọc là hàm số $h(\cdot)$ định nghĩa bởi:

$$h(x) = \sum_{l=0}^L h_l x^l \quad (2-18)$$

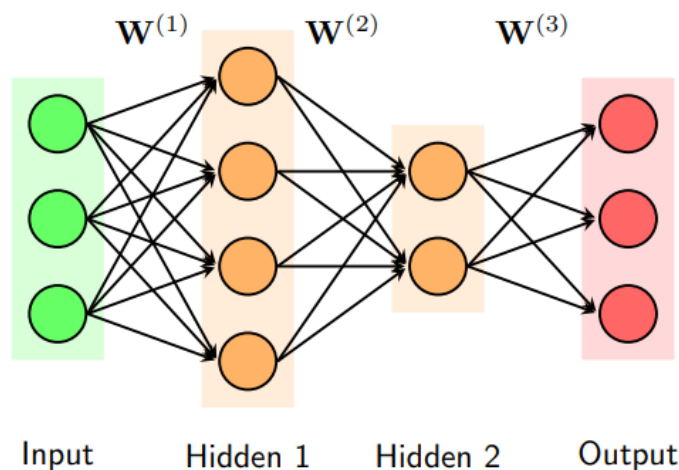
$$\Rightarrow \mathbf{H} = h(\mathbf{S}) \quad (2-19)$$

- Đáp ứng tần số của bộ lọc:

$$\begin{aligned}
 \tilde{\mathbf{h}} &= [\tilde{h}_n] = [h(\lambda_n)] \\
 > \mathbf{H} &= \sum_{l=0}^L h_l \mathbf{S}^l = \sum_{l=0}^L h_l \mathbf{V} \mathbf{\Lambda}^l \mathbf{V}^{-1} \\
 &= \mathbf{V} \left(\sum_{l=0}^L h_l \mathbf{\Lambda}^l \right) \mathbf{V}^{-1} \\
 &= \mathbf{V} h(\mathbf{\Lambda}) \mathbf{V}^{-1} = \mathbf{V} \text{diag}(\tilde{\mathbf{h}}) \mathbf{V}^{-1} \\
 \Rightarrow \tilde{y}_n &= \tilde{h}_n \tilde{x}_n
 \end{aligned} \tag{2-20}$$

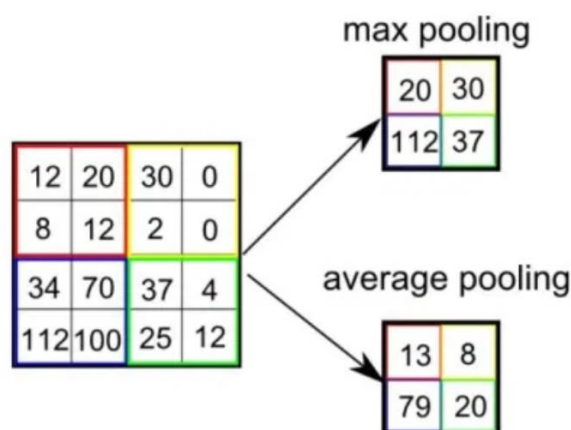
2.3 Mạng neural tích chập

- Mạng neural tích chập (Convolutional Neural Network - CNN) là thuật toán phổ biến của học sâu (Deep Learning) mang lại những cải tiến phát triển và nhanh chóng trong nhiều lĩnh vực, đặc biệt là thị giác máy tính (Computer Vision). Deep Learning dù chỉ là phần nhỏ trong Machine Learning nhưng đem đến những hiệu quả vượt bậc, nhờ sử dụng được ưu thế về dữ liệu [4].
- CNN giúp giải quyết các bài toán phân lớp đối tượng, hồi quy, chuyển đổi ngôn ngữ, ... thông qua việc học từ tập dữ liệu dùng để huấn luyện. Các dữ liệu ngõ vào sẽ thông qua các lớp (layer) để xử lý, lớp đầu tiên là input layer, lớp cuối cùng là output layer, các lớp trung gian là hidden layer. Trong CNN, phương pháp xử lý là tích chập (convolution) ngõ vào với một bộ lọc được xác định (kernel/ window/ mask/...) có chức năng phát hiện và tách đặc trưng của đối tượng, ngõ ra của lớp này sẽ là ngõ vào của lớp kế tiếp [11].



Hình 2-4: Mô hình mạng neural

- Số lượng, kích thước các bộ lọc càng đa dạng càng truy xuất được nhiều đặc trưng, nhưng lại gây phức tạp và nặng tính toán, CNN sẽ có lớp kết hợp (pooling layer) giữa các lớp tích chập để giảm kích thước của vector đặc trưng, giúp tránh hiện tượng quá khớp (overfitting) – mô hình khớp với dữ liệu huấn luyện nhưng lại không hoạt động tốt trên dữ liệu không nhìn thấy khi huấn luyện. Bộ lọc thường dùng cho lớp kết hợp có kích thước 2x2 thông qua 2 phương pháp cực đại hoặc trung bình.



Hình 2-5: Lớp kết hợp kích thước 2x2

- Qua các lớp trên, các thông tin sẽ được kết hợp tại lớp fully connected layer để đưa ra kết quả cuối cùng [12]. Các trọng số liên kết có thể nhận được thông qua thuật toán lan truyền ngược (backpropagation) được cập nhật từ các lớp phía sau đến lớp phía trước với mục tiêu tối thiểu hàm mất mát.
- Hàm mất mát (Loss Function) là cách đánh giá thuật toán mức độ của mô hình tốt như thế nào. Hàm mất mát cần phù hợp với điều kiện của riêng biệt cho từng bài toán cụ thể. Hàm mất mát thường dùng là ước lượng khả năng tối đa (maximum likelihood estimation) và Cross Entropy trong các bài toán phân lớp.

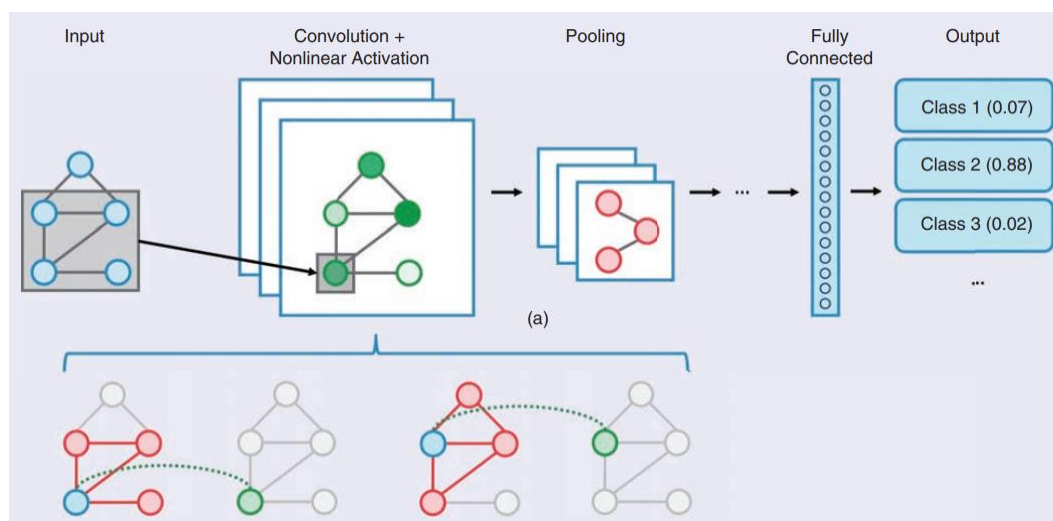
2.4 Graph CNN

- Thuật toán CNN hoạt động tốt với hình ảnh, có cấu trúc đồng nhất, nhưng nhiều ứng dụng với cấu trúc không đồng đều và đặc trưng bởi một cấu trúc hoặc đa tạp để mô hình như mạng xã hội, lưới giao thông, cấu trúc sinh học, ... CNN hoạt động không cần chú trọng đến ý nghĩa về cấu trúc của đối tượng, do đó bỏ lỡ nhiều đặc tính quan trọng và tính toán cồng kềnh do có quá nhiều sự liên kết là không cần thiết, hoặc không có nhiều ảnh hưởng. Trong khi đó Graph CNN (GCNN) quan tâm

đến các mối liên hệ của các đối tượng trong tập dữ liệu, do đó có thể giúp giảm số chiều cần tính toán mà vẫn giữ được các đặc trưng cơ bản.

- Cấu trúc tạo thành tương tự việc tạo sóng mang cho tín hiệu dữ liệu, nắm giữ kiến trúc cơ bản về mối quan hệ không gian giữa các đỉnh và cho phép nhiều các toán tử xử lý tín hiệu được thực hiện trên dữ liệu qua cấu trúc, chẳng hạn như lọc wavelet, tích chập và biến đổi Fourier. Kết hợp các kỹ thuật xử lý tín hiệu cấu trúc này với học sâu, có thể tìm hiểu trong các miền có khoảng cách không đều nhau (irregular).
- Các lớp trong GCNN có điểm tương đồng với CNN, từ tập dữ liệu ngõ vào \mathbf{x} và ngõ ra \mathbf{y} cho đồ thị đặc trưng bởi \mathbf{S} , ta cần xác định các hệ số bộ lọc để phép biến đổi $\mathbf{y} = \Phi(\mathbf{x}, \mathbf{S}, \mathcal{H})$ thỏa mãn điều kiện tối ưu.

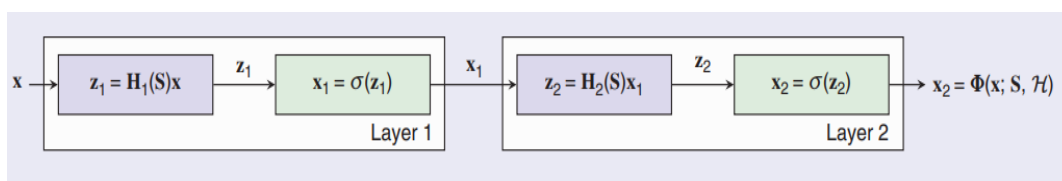
$$\mathcal{H}^* = \operatorname{argmin} \sum J(\Phi(\mathbf{x}, \mathbf{S}, \mathcal{H})) \quad (2-21)$$



Hình 2-6: Kiến trúc Graph CNN

- Trong đó, J là hàm sai số như MSE (mean squared error) hay cross-entropy. Phép biến đổi trong GCNN tương tự cũng là tích chập đồ thị với bộ lọc đa thức \mathbf{h} cho ra ngõ ra tại mỗi lớp là $\mathbf{H}\mathbf{x}$ [11]. Để tăng mức độ khớp dữ liệu thay vì chỉ dùng tích chập như một phép biến đổi tuyến tính, ta sẽ có thêm hàm phi tuyến $\sigma(\cdot)$.

$$\Phi(\mathbf{x}, \mathbf{S}, \mathcal{H}) = \sigma(\mathbf{H}\mathbf{x}) \quad (2-22)$$

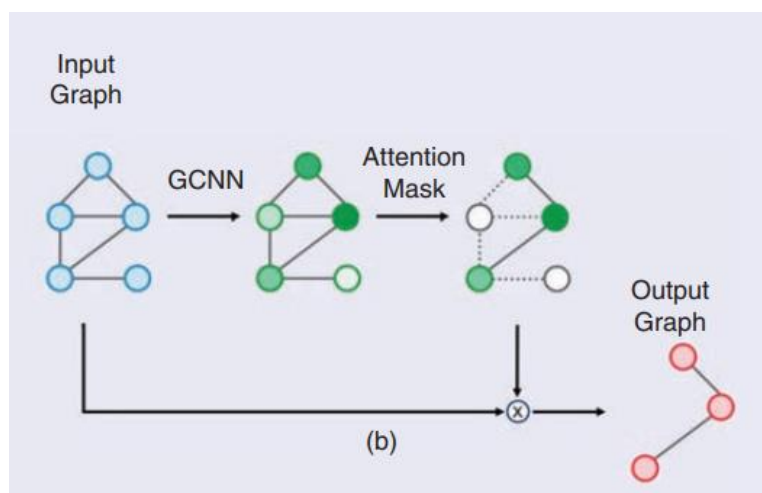


Hình 2-7: Mô hình GCNN với 2 lớp

- Bằng cách xây dựng L lớp nối tiếp với nhau như trong CNN ta cũng có được multilayer graph, trong đó mỗi lớp dữ liệu đồ thị tại lớp l được tính toán như sau:

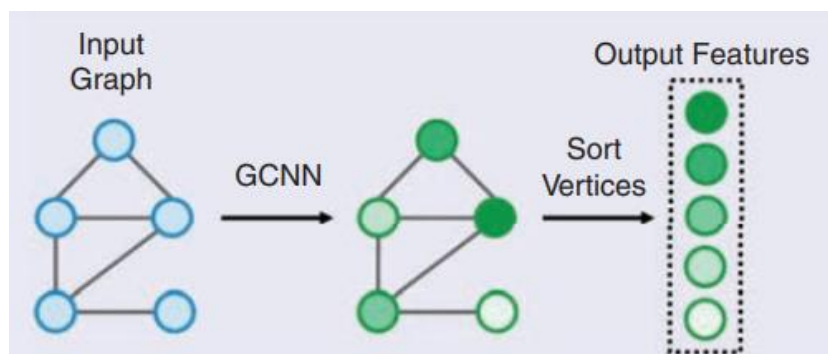
$$\mathbf{x}_l = \sigma(\mathbf{H}_l \mathbf{x}_{l-1}) \quad (2-23)$$

- Lớp tổng hợp (pooling layer) trong các mô hình phân loại cấu trúc để giảm số chiều và học tập theo thứ bậc. Tổng hợp cấu trúc các thuật toán thường làm giảm số lượng nút và thuật toán cũng thực thi biểu diễn phân cấp của dữ liệu, vì vậy GCNN có thể học quy mô lớn và các mẫu chung trong dữ liệu. Pooling là một hoạt động lấy mẫu xuống phi tuyến nhưng trong học sâu, không giống như trong DSP, khả năng khôi phục không phải là vấn đề đáng quan tâm chính đối với việc lấy mẫu xuống.



Hình 2-8: Tổng hợp cấu trúc phân biệt

- Trong CNN, tổng hợp tối đa thường được sử dụng, trong khi trong đồ thị CNN, không có sự nhất trí về cách tốt nhất để gộp các nút trong một biểu đồ. Các phương pháp tổng hợp đồ thị gần đây bao gồm Tổng hợp sắp xếp (SortPool), Tổng hợp phân biệt (DiffPool), Tổng hợp Top-k và Tổng hợp đồ thị Self-Attention (SagPool) [13].



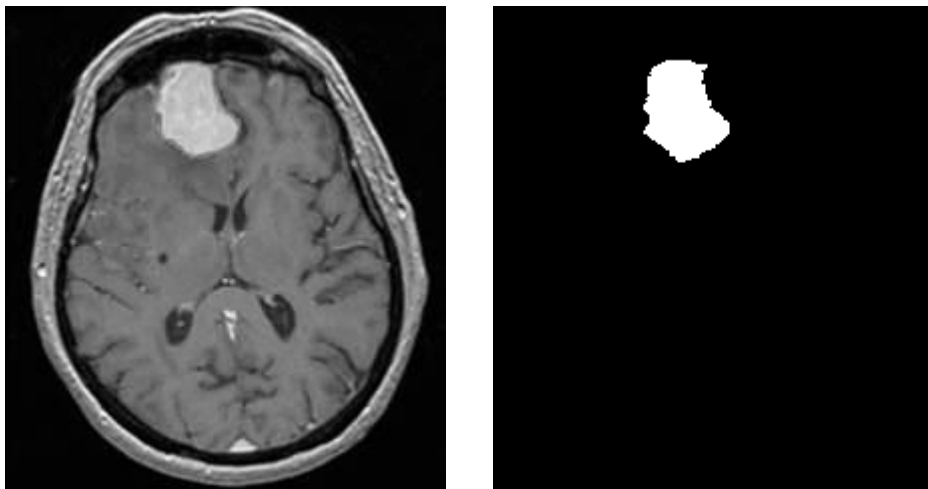
Hình 2-9: Tổng hợp cấu trúc sắp xếp

- Cuối cùng, ngõ ra được kết hợp tại lớp fully connected layer để đưa ra kết quả cuối cùng. Trong CNN thông thường, đầu vào thường có cùng kích thước và có thứ tự cố định. Tuy nhiên, trong các bài toán phân loại cấu trúc, người ta thường cần để xem xét các biểu đồ có nhiều kích thước khác nhau và với dữ liệu được xác định trên mỗi cấu trúc theo thứ tự hoán vị nút tùy ý. Các vector kết quả sẽ đóng vai trò là đầu vào cho một lớp được kết nối đầy đủ khác nhau về cả kích thước và thứ tự ghi nhãn, khiến cho việc so sánh trực tiếp trở nên khó khăn. Lớp cuối cùng, giải quyết vấn đề này bằng cách thu gọn các nút vào một số tính năng cố định, bất kể đầu vào là gì kích thước. Điều này thường được thực hiện sử dụng phép toán trung bình hoặc tổng trên tất cả các nút trong biểu đồ. Tuy nhiên, một số thống kê và phương pháp tiếp cận khác có thể được sử dụng như phổ cấu trúc khoảng cách hay sử dụng ma trận kề để nắm bắt thông tin toàn cục về cấu trúc biểu đồ bằng cách lấy khoảng cách điều hòa cho tất cả các nút.

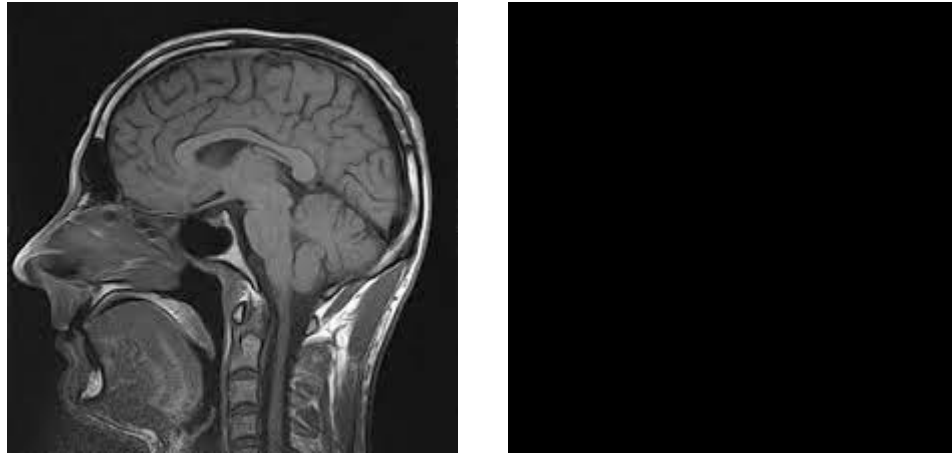
3. NỘI DUNG VÀ PHƯƠNG PHÁP NGHIÊN CỨU

3.1 Chuẩn bị tập dữ liệu

- Tập dữ liệu là các ảnh y khoa có cùng kích thước 50x50, được vector hóa thành các tín hiệu cấu trúc dạng vector cột, với mỗi pixel tương ứng một đỉnh trong cấu trúc. Trong tập dữ liệu chứa các ảnh y sinh có dấu hiệu bất thường (được gán nhãn xác định các vị trí bất thường đó) và những ảnh không có bất kỳ dấu hiệu bất thường nào. Tập dữ liệu huấn luyện (bao gồm ảnh cần xử lý và ảnh được gán nhãn) lần lượt chứa trong ma trận \mathbf{X} và \mathbf{Y} thành từng cặp tương ứng.
- Trong mục này, dữ liệu được lấy ngẫu nhiên từ các ảnh y sinh thực tế, do các ảnh có kích thước quá lớn nên thực hiện trên từng phần miền nhỏ trong ảnh. Vector \mathbf{x}_i là các cột của ma trận \mathbf{X} , ứng với ảnh thứ i trong tập dữ liệu, \mathbf{y}_i tương ứng được gán nhãn làm nổi bật vùng bất thường bằng tín hiệu nhị phân, bằng 1 tương đương với điểm có bất thường, ngược lại pixel có giá trị bằng 0 là điểm không có dấu hiệu bất thường.



Hình 3-1: Ảnh chụp chứa bất thường trong não



Hình 3-2: Ảnh chụp não không có bất thường

3.2 Xây dựng cấu trúc ứng dụng khảo sát ảnh y khoa

- Tiếp theo, cần xác định mô hình cấu trúc đại diện (graph representation) phù hợp với mục tiêu. Vấn đề lựa chọn mô hình được thực hiện thông qua học học đồ hình (graph learning) với tập dữ liệu sao cho phù hợp nhất, thể hiện được mối liên hệ giữa các đỉnh của tập dữ liệu.
- Với tập dữ liệu \mathbf{X} được thu thập từ các ảnh y khoa và \mathbf{X}_i là một tín hiệu cấu trúc trong \mathbf{X} . Các phương pháp thực hiện tính toán có thể được chuẩn hóa và giảm bớt các liên kết yếu (tùy mức độ cần chính xác) bằng cách đặt ra mức ngưỡng liên kết.

$$A_{ij_thres} = \begin{cases} A_{ij}, & A_{ij} \geq A_{thres} \\ 0, & A_{ij} < A_{thres} \end{cases} \quad (3-1)$$

- Có nhiều phương pháp để lựa chọn cấu trúc phù hợp (có thể sử dụng cấu trúc cố định hoặc linh động tùy theo ngõ vào hoặc lớp thực hiện tích chập) như dựa vào khoảng cách Euclide, khoảng cách Gaussian hay tương quan giữa các đỉnh của tập dữ liệu và nhiều phương pháp kết hợp khác. Ví dụ, phương pháp khoảng cách Gaussian với độ lệch chuẩn σ :

$$A_{ij} = \exp\left(-\frac{\|\mathbf{X}_i - \mathbf{X}_j\|^2}{2\sigma^2}\right) \quad (3-2)$$

- Cấu trúc sử dụng trong phần này tương tự với ảnh thông thường trong CNN. Điểm khác biệt chính là ngoài việc xem xét các điểm lân cận còn quan tâm đến mức xám để xác định các liên kết chính yếu, loại bỏ các liên kết yếu ít ảnh hưởng đến kết quả thông qua mức ngưỡng (3-1). Ma trận kề tạo ra thông qua khoảng cách Euclid và sự tương đồng về mức xám suy giảm theo hàm mũ.

- Liên kết chặt nhất là 1, ứng với pixel liền kề và cùng mức xám, ngược lại với liên kết yếu. Các liên kết yếu được lấy ngưỡng loại bỏ xem như không có mối quan hệ khi suy giảm 10%.

```
%modelize graph representation
for i = 1: N
    for j = 1:N
        ui = rem(i,d1); if ui == 0;ui = d1; end
        vi = (i-ui)/d1+1;
        uj = rem(j,d1); if uj == 0;uj = d1; end
        vj = (j-uj)/d1+1;

        dis = sqrt((ui-uj)^2+(vi-vj)^2);
        dif = abs(img(ui,vi)-img(uj,vj));
        A(i,j) = exp(-dis-dif+1);
        if A(i,j) < 0.1; A(i,j) = 0; end
    end
end
```

3.3 Xác định hàm truyền thuận

- Các lớp trong mạng bao gồm phần tuyến tính thực hiện bởi phép nhân ma trận trọng số hay tích chập và phần phi tuyến thực hiện bởi hàm kích hoạt hay pooling. Phần này các hàm kích hoạt sử dụng là ReLU (Rectified Linear Unit – chỉnh lưu tuyến tính) với công thức cụ thể như sau:

$$ReLU(z) = \max\{0, z\} = z \cdot u(z)$$

$$\frac{d}{dz} ReLU(z) = u(z) = \begin{cases} 0, & x \leq 0 \\ 1, & x > 0 \end{cases} \quad (3-3)$$

Với $u(\cdot)$ là hàm đơn vị, hàm kích hoạt ReLU cho đạo hàm đơn giản, tính toán nhanh chóng và là hàm đặt lên từng phần tử (element-wise). Riêng lớp cuối cùng được kích hoạt bằng hàm sigmoid (cũng là hàm element-wise) như sau:

$$sigmoid(z) = \frac{1}{1 + e^{-z}} = \frac{e^z}{e^z + 1}$$

$$\frac{d}{dz} sigmoid(z) = \frac{e^{-z}}{(1 + e^{-z})^2} = sigmoid(z)(1 - sigmoid(z)) \quad (3-4)$$

- Phần tuyến tính trong mạng sẽ thay tích chập trong không gian 2 chiều bằng phép chập cấu trúc với toán tử \mathbf{S} , cho tín hiệu \mathbf{x} sẽ thu được tín hiệu \mathbf{y} theo công thức (2-17). Với bộ lọc bậc nhất có hai hệ số bộ lọc bằng nhau được chứng tỏ là hoạt động tốt, giúp tổng quát hóa dữ liệu và đơn giản trong tính toán:

$$\begin{aligned}
 \mathbf{y} &= \mathbf{H}\mathbf{x} \\
 \mathbf{H} &= \sum_{l=0}^1 h_l \mathbf{S}^l = h_0 \mathbf{I} + h_1 \mathbf{S} = w(\mathbf{I} + \mathbf{S}) \\
 \Rightarrow \mathbf{y} &= (\mathbf{I} + \mathbf{S})\mathbf{x}w
 \end{aligned} \tag{3-5}$$

- Trong đó toán tử \mathbf{S} được chọn là ma trận kề chuẩn hóa, khi có F_1 tín hiệu ngõ vào và F_0 tín hiệu ngõ ra, ta biểu diễn thành dạng ma trận như sau:

$$\begin{aligned}
 \mathbf{Y} &= (\mathbf{I} + \mathbf{S})\mathbf{X}\mathbf{W} = \hat{\mathbf{A}}\mathbf{X}\mathbf{W} \\
 \hat{\mathbf{A}} &= \mathbf{I} + \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}
 \end{aligned} \tag{3-6}$$

- Hàm truyền thuận của mạng có L lớp với \mathbf{X} ngõ vào và $\hat{\mathbf{Y}}$ là ngõ ra của mạng, được biểu diễn như sau:

$$\begin{aligned}
 \mathbf{A}^1 &= \mathbf{X} \\
 \mathbf{Z}^l &= \hat{\mathbf{A}}\mathbf{A}^l\mathbf{W}, \quad l = 1, 2, \dots, L \\
 \mathbf{A}^l &= \text{ReLU}(\mathbf{Z}^{l-1}), \quad l = 2, 3, \dots, L \\
 \hat{\mathbf{Y}} &= \text{sigmoid}(\mathbf{Z}^L)
 \end{aligned} \tag{3-7}$$

- Đoạn code dưới đây thực hiện hàm truyền thuận:

```

%initial
W = cell(1,L);
Z = cell(1,L); A = cell(1,L);
E = cell(1,L);
for layer = 1:L
    W{layer} = randn(Fx(layer),Fx(layer+1));
    b{layer} = randn(Fx(layer+1),1);
end

%forward
A{1} = X;
for layer = 1:L
    H = A{layer};
    for ind = 1:nd
        temp = reshape(H(:,ind),N,F(layer));
        temp = Ad*temp;
        H(:,ind) = reshape(temp,Fx(layer),1);
    end
    A{layer} = H;
    Z{layer} = W{layer}'*H;
    if layer == L; break; end
    A{layer+1} = max(0,Z{layer});
end
Yest = 1./(1+exp(-Z{L}));

```

3.4 Xây dựng hàm mất mát và lan truyền ngược

- Cuối cùng là xác định hàm mất mát và cập nhật trọng số bằng gradient descent theo cơ chế lan truyền ngược. Việc đào tạo thông qua đường truyền thuận để có được kết

quả đầu ra, với sai số được lan truyền ngược thông qua mạng để cập nhật các trọng số [14]. Việc cập nhật mất nhiều thời gian tính toán đến khi các trọng số ổn định.

- Hàm mất được sử dụng là cross-entropy, với $\hat{\mathbf{y}}$ là ngõ ra của mạng và \mathbf{y} là nhãn thì hàm mất mát định nghĩa như sau:

$$J = -\frac{1}{nd} \sum_{j=1}^{nd} \sum_{i=1}^N (y_{ij} \log(\hat{y}_{ij}) + (1 - y_{ij}) \log(1 - \hat{y}_{ij})) \quad (3-8)$$

- Trong đó, nd là số lượng dữ liệu huấn luyện, N là số chiều vector nhãn \mathbf{y} (hay số pixel trong ảnh ngõ vào). Tính toán gradient của hàm mất mát theo trọng số cho 1 dữ liệu để tối ưu hóa hàm mất mát bằng gradient descent:

$$\begin{aligned} \frac{\partial J}{\partial w_{ij}^l} &= \sum_k \frac{\partial J}{\partial z_{kj}^l} \frac{\partial z_{kj}^l}{\partial w_{ij}^l} = \sum_k e_{kj}^l \cdot (\hat{\mathbf{A}} \mathbf{A}^l)_{ki} \\ \nabla_{\mathbf{W}^l} J &= (\hat{\mathbf{A}} \mathbf{A}^l)^T \mathbf{E}^l = \mathbf{A}^{lT} \hat{\mathbf{A}} \mathbf{E}^l \end{aligned} \quad (3-9)$$

- Thực hiện tính toán các thành phần ma trận \mathbf{E} :

$$\begin{aligned} e_{kj}^l &= \frac{\partial J}{\partial z_{kj}^l} = \frac{\partial J}{\partial a_{kj}^{l+1}} \frac{\partial a_{kj}^{l+1}}{\partial z_{kj}^l} = \frac{\partial J}{\partial a_{kj}^{l+1}} \cdot u(z_{kj}^l) \\ &= u(z_{kj}^l) \cdot \sum_m \sum_n \frac{\partial J}{\partial z_{mn}^{l+1}} \frac{\partial z_{mn}^{l+1}}{\partial a_{kj}^{l+1}} \\ &= u(z_{kj}^l) \cdot \sum_m \sum_n e_{mn}^{l+1} \frac{\partial z_{mn}^{l+1}}{\partial a_{kj}^{l+1}} \\ &= u(z_{kj}^l) \cdot \sum_m \sum_n e_{mn}^{l+1} \hat{A}_{mk} w_{jn}^{l+1} \\ \mathbf{E}^l &= \hat{\mathbf{A}}^T \mathbf{E}^{l+1} \mathbf{W}^{l+1T} \odot u(\mathbf{Z}^l), \quad l = 1, 2, \dots, L-1 \end{aligned} \quad (3-10)$$

- Trong đó, \odot là phép nhân nhân từng thành phần (element-wise product). Ma trận sai số lớp cuối cùng được tính toán trực tiếp từ hàm mất mát và hàm kích hoạt lớp cuối cùng:

$$\begin{aligned} e_k^l &= \frac{\partial J}{\partial z_k^l} = \frac{\partial J}{\partial \hat{y}_k} \frac{\partial \hat{y}_k}{\partial z_k^l} \\ &= -\frac{\partial}{\partial \hat{y}_k} (y_k \log(\hat{y}_k) + (1 - y_k) \log(1 - \hat{y}_k)) \cdot \frac{\partial}{\partial z_k^l} \text{sigmoid}(z_k^l) \\ &= -\left(\frac{y_k}{\hat{y}_k} - \frac{1 - y_k}{1 - \hat{y}_k} \right) \cdot \text{sigmoid}(z_k^l) (1 - \text{sigmoid}(z_k^l)) \\ &= -\left(\frac{y_k}{\hat{y}_k} - \frac{1 - y_k}{1 - \hat{y}_k} \right) \cdot \hat{y}_k (1 - \hat{y}_k) \\ &= -(y_k (1 - \hat{y}_k) - (1 - y_k) \hat{y}_k) \\ &= \hat{y}_k - y_k \end{aligned} \quad (3-11)$$

- Quá trình lan truyền ngược được tiến hành từ lớp cuối L đến lớp đầu tiên, biểu diễn cụ thể dưới dạng ma trận như sau (trong đó η là tốc độ học) như sau:

$$\begin{aligned}
 \mathbf{E}^L &= \hat{\mathbf{Y}} - \mathbf{Y} \\
 \mathbf{E}^l &= \hat{\mathbf{A}}^T \mathbf{E}^{l+1} \mathbf{W}^{l+1T} \odot u(\mathbf{Z}^l), \quad l = 1, 2, \dots, L-1 \\
 \nabla_{\mathbf{W}^l} J &= (\hat{\mathbf{A}} \mathbf{A}^l)^T \mathbf{E}^l = \mathbf{A}^{lT} \hat{\mathbf{A}} \mathbf{E}^l, \quad l = 1, 2, \dots, L \\
 \mathbf{W}^l &\leftarrow \mathbf{W}^l - \eta \nabla_{\mathbf{W}^l} J, \quad l = 1, 2, \dots, L
 \end{aligned} \tag{3-12}$$

- Đoạn code mô tả quá trình cập nhật vừa nêu:

```

%backward
E{L} = (Yest - Y)/nd/N;
for layer = linspace(L,1,L)
    gW = A{layer}*E{layer}';
    W{layer} = W{layer} - rate*gW;

    if layer == 1; break; end
    Adx = zeros(Fx(layer));
    for iF = 1:F(layer)/N;
        Adx((iF-1)*N+1:iF*N, (iF-1)*N+1:iF*N) = Ad; end

    difAc = (Z{layer-1}>0);
    E{layer-1} = Adx*W{layer}*E{layer}.*difAc;
end

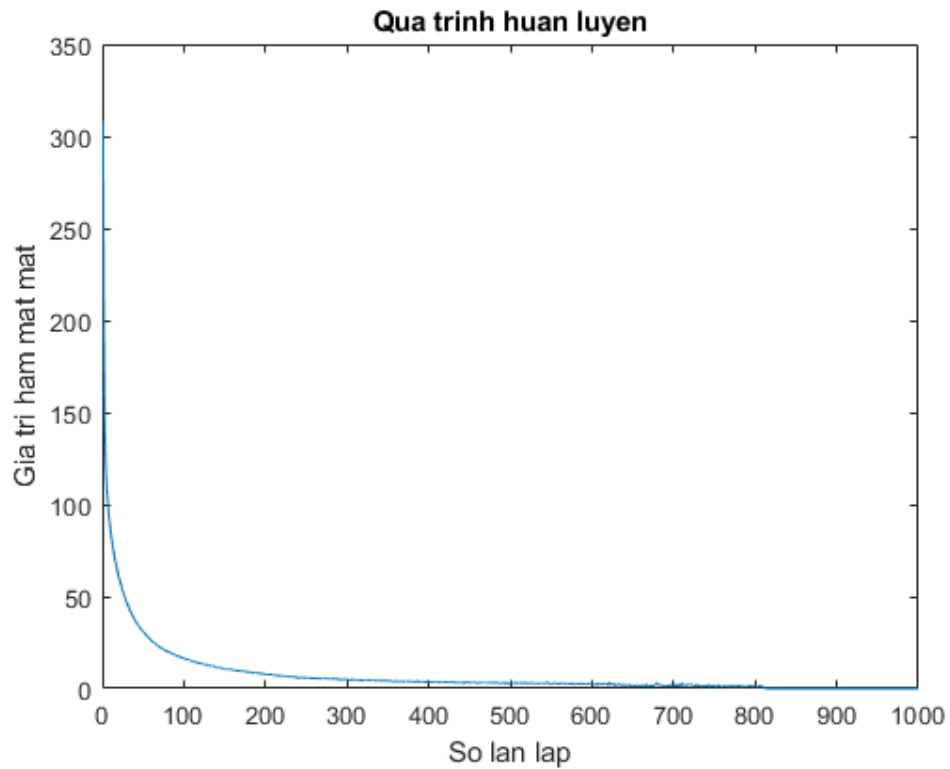
```

3.5 Tiến hành huấn luyện và đánh giá kết quả

- Từ những tính toán trên, tiến hành huấn luyện mạng với tập dữ liệu, do dữ liệu thử nghiệm tương đối nhỏ nên thực hiện đồng loạt (batch) trên tập dữ liệu với cả ma trận dữ liệu \mathbf{X} .
- Thử nghiệm và đánh giá mô hình với các dữ liệu trong và ngoài tập dữ liệu để có kết quả khách quan. Tính toán sai số trên tập dữ liệu thực hiện huấn luyện để biết khả năng của mô hình mạng có cấu trúc.
- Nhận xét hiệu quả mô hình đã khảo sát trên ảnh y khoa cho việc phân tích các bất thường.

4. KẾT QUẢ THỰC HIỆN

- Sau khi tiến hành huấn luyện, kết quả cho sai số rất bé với tập dữ liệu, nhưng tốc độ tính toán hội tụ chậm, tùy thuộc vào việc lựa chọn và tận dụng tốc độ học. Nguyên nhân làm chậm hội tụ do quá trình xử lý đồng loạt trên toàn bộ tập dữ liệu. Kết quả thực hiện:

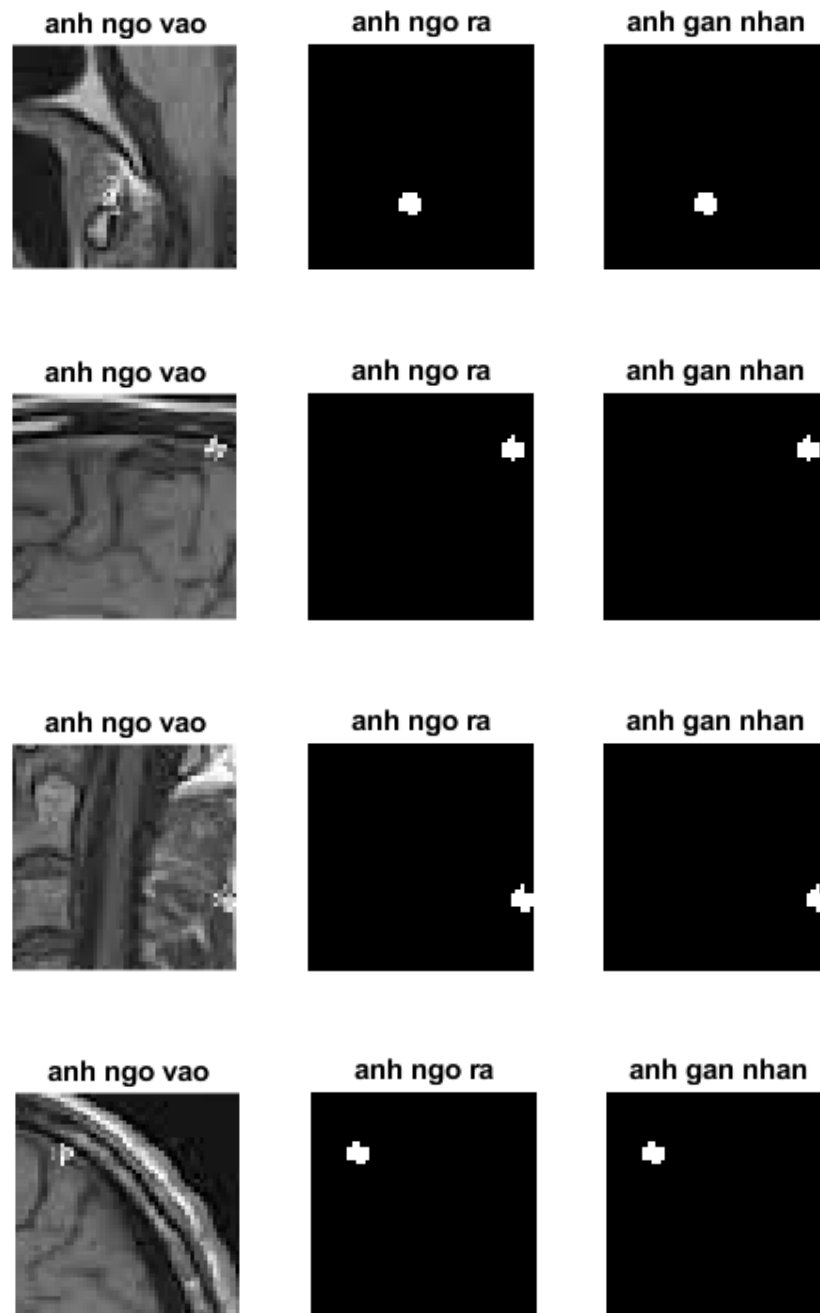


Hình 4-1: Quá trình tối ưu sai số

- Một số ảnh kết quả giữa ngõ ra mạng và nhãn cho thấy hiệu quả rất cao, có tiềm năng lớn.

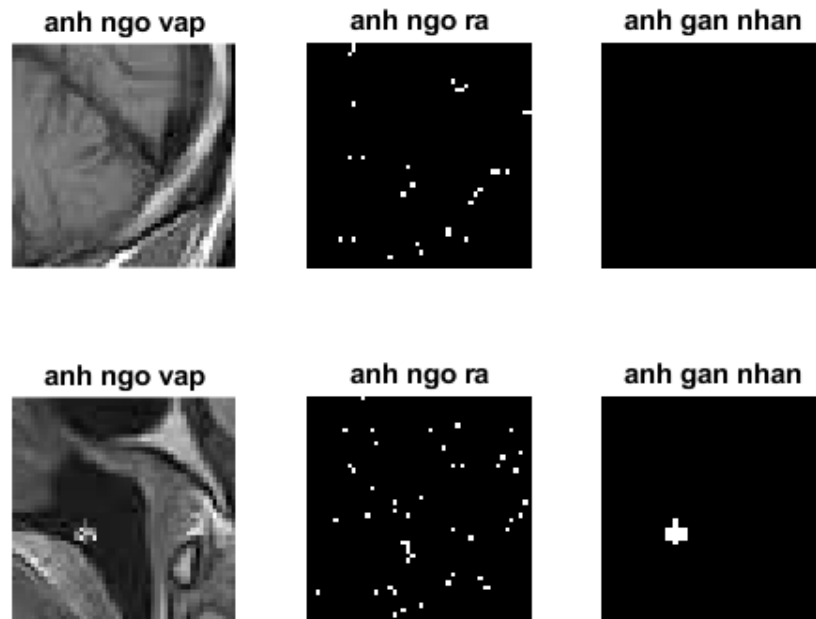


Hình 4-2: Kết quả ảnh không có bất thường



Hình 4-3: Kết quả ngõ ra sau khi huấn luyện

- Dễ thấy, kết quả phù hợp với bộ dữ liệu gồm 100 ảnh, do đó không hy sinh dữ liệu nhiều, sai số gần như tiến về 0. Tuy nhiên, việc sai số quá thấp cũng dẫn đến khả năng mô hình bị quá khớp (overfitting) với tập dữ liệu. Cụ thể, những kết quả tính toán sau cho dữ liệu ngoài tập dữ liệu đã không chính xác.



Hình 4-4: Các kết quả bị quá khớp

- Kết quả trên bị khớp quá nhiều, không thực hiện tốt trên dữ liệu chưa nhìn thấy, đọc sai các vùng bất thường và nghiêm trọng hơn là các ảnh không có bất thường vẫn tìm ra vùng bất thường. Nguyên nhân có thể do thực hiện chưa tốt phân phi tuyến trong mạng, dữ liệu chưa được tổng quát hóa tối đa. Hơn hết, dữ liệu ở quy mô thử nghiệm nhỏ, không đủ tính bao quát để mạng có thể học tập sâu hơn.

5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết luận

- Nội dung đề án đã thực hiện tìm hiểu một khái niệm tương đối mới: “cấu trúc” (graph) và có tiềm năng phát triển lớn. Những kết quả thực hiện khảo cho thấy ứng dụng khái niệm graph trong huấn luyện mạng có nhiều lợi điểm hơn so với phương pháp học truyền thống qua cả lý thuyết và các thử nghiệm với quy mô nhỏ trên MatLab.
- Các khái niệm đã được tìm hiểu và trình bày như: tín hiệu cấu trúc, phổ cấu trúc, bộ lọc cấu trúc, ... sử dụng để mô hình hóa trong ảnh y sinh tăng mức độ liên kết giữa nội dung tập dữ liệu.
- Việc mô hình mạng có cấu trúc gặp phải vấn đề khó khăn ngay khi bắt đầu là lựa chọn mô hình cấu trúc phù hợp. Việc này không tồn tại khi giải quyết bằng các phương pháp truyền thống với cấu trúc đồng đều. Khi cấu trúc trong mô hình mạng lựa chọn là đồng đều thì ta lại thu được mô hình mạng cơ bản như đã biết.
- Phương pháp GCNN về lý thuyết cũng sẽ gây nặng nề hơn cho tính toán khi phải lưu trữ và xử lý trên cấu trúc, xuất hiện nhiều ma trận mới cần giải quyết, sẽ phức tạp nếu bậc bộ lọc cấu trúc cao, đặc biệt là tính toán lan truyền ngược.
- Kết quả mô hình mạng sử dụng cấu trúc bước đầu cho hiệu quả tốt, được kỳ vọng có hiệu quả cao hơn so với các phương pháp trước đó như CNN, RNN, SVM, ... do tận dụng được đặc điểm về cấu trúc riêng biệt trên ảnh y sinh, giúp phân tách và phát hiện bất thường của đối tượng khảo sát. Có ý nghĩa thực tiễn giúp giảm bớt gánh nặng công việc cho các bác sĩ, sàng lọc sớm các trường hợp nghi ngờ có vấn đề bất thường.

5.2 Hướng phát triển

- Hạn chế về hàm phi tuyến ReLU quá đơn giản cần, do đó cần tìm ra hàm phi tuyến phù hợp hơn cho mạng thông qua các thử nghiệm khác nhau, tăng cường sự tổng quát hóa, đồng thời tránh overfitting.
- Tập dữ liệu triển khai trên nguồn ảnh y sinh lớn hơn chứa với nhiều vùng bất thường thực tế, trên những cơ quan khác trên cơ thể. Đồng thời có khả năng phát hiện và nhận biết cả bất thường đó là gì, mức độ ảnh hưởng như thế nào.

- Tìm hiểu về việc sử dụng linh hoạt cả miền không gian cấu trúc và miền phổ. Các bài báo gần đây cũng chứng tỏ việc xử lý trên miền không gian mang lại hiệu quả tốt hơn. Tuy nhiên việc thực hiện chập trên miền không gian nặng nề về phương diện tính toán hơn so với nhân từng phần tử ở miền tần số.
- Thực hiện trên các mô hình cấu trúc khác nhau để đánh giá các kết quả của từng mô hình mang lại. Từ đó, lựa chọn ra cấu trúc phù hợp nhất, có thể biến đổi linh hoạt để tăng chất lượng
- Vấn đề quá khớp chưa giải quyết, do đó cần tìm hiểu thêm để hạn chế quá khớp, giúp mô hình hoạt động hiệu quả hơn.

6. TÀI LIỆU THAM KHẢO

- [1] M. Newman, *Networks: an introduction.*, Oxford University Press, 2010.
- [2] S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular," *IEEE Signal Process. Mag.*, vol. 30, pp. 83-98, May 2013.
- [3] Leo Grady and Jonathan R. Polimeni, *Discrete Calculus - Applied Analysis on Graphs*, Springer, 2010.
- [4] Ronneberger, O., Fischer, P., and Brox, T., "U-net: convolutional networks for biomedical image segmentation," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015.
- [5] Katkar, J. A. and Baraskar, T., "Medical image segmentation using PCA and k-mean clustering algorithm," in *Post Graduate Conference for Information Technology*, 2016.
- [6] Edwards, M. and Xie, X., "Graph based convolutional neural network," in *British Machine Vision Conference*, 2016.
- [7] Bahade, S., Edwards, M. and Xie, X., "Graph Convolution Networks for Cell Segmentation," in *Proceedings of the 10th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2021)*, 2021.
- [8] C. v. Tycowicz, "Towards shape-based knee osteoarthritis classification using graph convolutional networks," 2019.

- [9] A. Ortega, P. Frossard, J. M. F. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proc. IEEE*, vol. 106, p. 808–828, May 2018.
- [10] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun, "Spectral networks and locally connected networks on graphs," *CoRR*, *abs/1312.6203*, 2013.
- [11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit (CVPR)*, pp. 1-9, Jun. 2016.
- [12] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit (CVPR)*, p. 3431–3440., Jun. 2015.
- [13] Chevalier, C. and Safro, "Comparison of coarsening schemes for multilevel graph partitioning," *International Conference on Learning and Intelligent Optimization*, vol. I, p. 191–205, 2009.
- [14] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, Vols. 323, no. 6088, p. 533–536, Oct. 1986.