Trường Đại học Khoa học tự nhiên

Khoa Công nghệ thông tin
26/03/2017

Báo cáo đồ án

Kĩ thuật trí tuệ nhân tạo:

Mạng nơron trong nhận dạng âm thanh

Thực hiện: 1412630 – Đỗ Khánh Long Tường

MỤC LỤC

I. Thông tin cá nhân

STT	MSSV	Họ tên	Email
1	1412630	Đỗ Khánh Long Tường	hanhdospla@gmail.com

II. Giới thiệu chủ đề và động lực tìm hiểu

1. Giới thiệu chủ đề

Hiện nay với sự phát triển của khoa học máy tính, ngày càng có nhiều nghiên cứu được thực hiện nhằm tận dụng tối đa khả năng của máy tính để hỗ trợ con người thực hiện những công việc mà con người không thể hoàn thành hoặc hoàn thành không tốt. Một trong số những vấn đề dạng trên là việc nhận dạng các loại âm thanh khác nhau trong các môi trường áp dụng vào nhiều lĩnh vực như y tế, anh ninh, nghệ thuật,…

Nhận dạng âm thanh có thể được sử dụng trong nhận diện đối tượng tương tự như bài toán theo vết người, từ đó có thể áp dụng vào truy vết đối tượng trong an ninh, đếm số người dựa vào tần số và cường độ âm thanh bước chân trong các nhà hàng, khách sạn,…Bên cạnh đó, so với việc lắp đặt một hệ thống xử lý ảnh, việc lắp đặt hệ thống xử lý tín hiệu âm đơn giản hơn rất nhiều, tiết kiệm về chi phí và giải quyết được vấn đề lưu trữ.

Mặc dù thực tế xã hội cần rất nhiều hệ thống xử lý âm thanh, tuy nhiên vấn đề này vẫn chưa được thực sự quan tâm đúng mức. Đó chính là lý do chính mà chủ đề "Nhận diện âm thanh" được lựa chọn để đưa vào thực hiện đồ án.

2. Động lực tìm hiểu:

2.1 Về mặt khoa học:

- Khi tìm hiểu về chủ đề nhận dạng âm thanh ta sẽ cập nhật được nhiều kiến thức mới như sound understanding cho phép ta hiểu hơn về cơ chế làm việc và hoạt động của âm thanh dựa vào tín hiệu hay machine learning cung cấp cho ta một lượng lớn kiến thức về mảng huấn luyện hệ thống và cung cấp AI cho hệ thống (cụ thể ở đây là Neural Network),… Các kiên thức này có nhiều áp dụng khác nhau trên nhiều lĩnh vực và vì thế khi nghiên cứu những đề tài khác, ta có thể tiết kiệm được thời gian tìm hiểu lại những vấn đề này.
- Tìm hiểu chủ đề cho phép chúng nắm được cách tư duy cũng như phương pháp luận khác nhau của các tác giả đã từng nghiên cứu cùng lĩnh vực. Từ đó, người tìm hiểu có thể rút ra cho mình những bài học kinh nghiệm cũng như đưa ra những ý tưởng mới nhằm cải tiến hệ thống sao cho phù hợp.

2.2 Về mặt ứng dụng:

Chủ đề nhận dạng âm thanh đang là một chủ đề không mới, được các nghiên cứu sinh cũng như các nhà khoa học tham gia tìm hiểu vì nó đem lại cho cuộc sống nhiều ứng dụng trên các lĩnh vực khác nhau. Sau đây là một vài ứng dụng tiêu biểu mà chúng ta có thể thực hiện được khi tìm hiểu chủ đề nhận dạng âm thanh:

- Rút trích âm thanh trong một đoạn video, hội thoại và tiến hành phân tích các loại âm tần, từ đó xác định ra nguồn phát âm có trong video, đoạn hội thoại đó. Ứng dụng này được sử dụng nhiều trong an ninh và tìm kiếm tội phạm dựa vào thông tin nhận được từ các nguồn phát âm đã nhận dạng, cảnh sát có thể dễ dàng khoanh vùng đối tượng trong quá trình tìm kiếm và rà soát.
- Nhận dạng tiếng nói là một hệ thống được ứng dụng khá thành công trên các thiết bị di động, nó cho phép đối thoại giữa người dùng và thiết bị đồng thời cho phép người dùng thực hiện các chức năng trên điện thoại thông qua ngôn ngữ nói.
- Nhận dạng âm thanh có thể áp dụng trong lĩnh vực y học giúp phát hiện âm dị thường của các cơ quan nội tạng trong cơ thể. Với hệ thống này, một máy phát sóng sẽ phóng các sóng âm lan truyền trong cơ thể, sóng âm sẽ phản xạ lại khi tiếp xúc với bề mặt bất kì trong cơ quan nội tạng, dựa vào thời gian nhận tín hiệu sóng, dạng sóng, cường độ, cao độ mà ta có thể nhận biết các tế bào, cơ quan, nội tạng trong cơ thể có đang ở trạng thái bình thường không.
- ...

Với các ứng dụng trên, ta có thể thấy được một tiềm năng lớn của nhận dạng âm thanh trong cải thiện cuộc sống con người, rất đáng để tìm hiểu và nghiên cứu.

III. Các nội dung tìm hiểu

1. Giới thiệu chung:

a. Giới thiệu về âm thanh:

Định nghĩa âm thanh:

- Âm thanh là các dao động cơ học (biến đổi vị trí qua lại) của các phân tử, nguyên tử hay các hạt làm nên vật chất và lan truyền trong vật chất như các sóng. Âm thanh, giống như nhiều sóng, được đặc trưng bởi tần số, bước sóng, chu kỳ, biên độ và vận tốc lan truyền (tốc độ âm thanh).
- Đối với thính giác của người, âm thanh thường là sự dao động, trong dải tần số từ khoảng 20 Hz đến khoảng 20000 Hz, của các phân tử không khí, và lan truyền trong không khí, va đập vào màng nhĩ, làm rung màng nhĩ và kích thích bộ não. Tuy nhiên âm thanh có thể được định nghĩa rộng hơn, tuỳ vào ứng dụng, bao gồm các tần số cao hơn hay thấp hơn tần số mà tai người có thể nghe thấy, không chỉ lan truyền trong không khí mà còn truyền trong bất cứ vật liệu nào. Trong định nghĩa rộng này, âm thanh là sóng cơ học và theo lưỡng tính sóng hạt của vật chất, sóng này cũng có thể coi là dòng lan truyền của các hạt phonon, các hạt lượng tử của âm thanh.
- Cả tiếng ồn và âm nhạc đều là các âm thanh. Trong việc truyền tín hiệu bằng âm thanh, tiếng ồn là các dao động ngẫu nhiên không mang tín hiệu.

Tính chất âm thanh:

- Âm thanh mang tính chất của sóng dọc, tức là nó là sự lan truyền dao động của đại lượng vô hướng là áp suất, đồng thời là sự lan truyền dao động của đại lượng có hướng là vận tốc và vị trí của các phân tử hay nguyên tử trong môi trường, trong đó phương dao động luôn trùng với phương chuyển động của sóng.
- Cũng như các sóng cơ học khác, sóng âm mang năng lượng tỉ lệ với bình phương biên độ sóng. Năng lượng đó truyền đi từ nguồn âm đến tai ta. Cường độ âm thanh là lượng năng lượng được sóng âm truyền đi trong một đơn vị thời gian qua một đơn vị diện tích đặt vuông góc với phương truyền âm. Ngoài ra trường độ cũng góp phần ảnh hưởng đến chất lượng âm thanh.

Biểu diễn tín hiệu âm:

- **Tín hiệu sóng (Waveplot):** cho thông tin về tần số bước sóng của âm thanh. [1]
- **Phổ đồ (Spectrogram):** cho thông tin về tầng số và thời gian của âm thanh thông qua phép biến đổi STFT (Short-time Fourier Transform).[1]
- **Log Power Spectrogram**: là phổ đồ Gaussian Kernel có độ rộng thay đổi theo hàm log.[1]

Phép biến đổi Short-time Fourier (Short-time Fourier Transform – STFT) [7]:

- Ta có tín hiệu nguồn trong miền thời gian là $f(t)$. Với tín hiệu này, ta có thông tin đầy đủ về mặt thời gian (tại thời điểm t thì mức năng lượng tương ứng là $f(t)$).
- Biến đổi Fourier của tín hiệu này là $F(\omega)$ cho ta đầy đủ thông tin trong miền tần số, nhưng hoàn toàn không có thông tin gì về miền thời gian (với tần số ω thì độ lớn của nó trong tín hiệu là $F(\omega)$, nhưng ta không biết là tần số này xuất hiện vào thời điểm nào trong miền thời gian). Như vậy sau biến đổi Fourier, ta có thông tin trong miền tần số nhưng mất hoàn toàn thông tin về miền thời gian.
- Đây chính là lúc mà Short-time Fourier transform (STFT) xuất hiện. Ý tưởng chính của STFT là "hi sinh" một ít thông tin về các tần số thấp trong miền tần số để có thêm thông tin về miền thời gian. STFT được biểu diễn bằng 1 hàm $G(\omega, t)$ theo 2 biến là tần số ω và thời gian t. Như vậy nhìn vào kết quả của STFT, ta có thể biết là tần số ω xuất hiện vào thời điểm nào trong miền thời gian.
- STFT phổ biến nhất là Gabor transform. Trong Gabor transform, ta chọn 1 hàm kernel (thông dụng nhất có lẽ là hàm Gaussian, trong xử lí tín hiệu số người ta gọi là filter), rồi trượt hàm kernel này trong miền thời gian, thực hiện phép tích chập, rồi biến đổi Fourier. Như vậy mỗi filter được đặc trưng bằng 2 tham số là vị trí và độ rộng của filter đó trong miền thời gian.

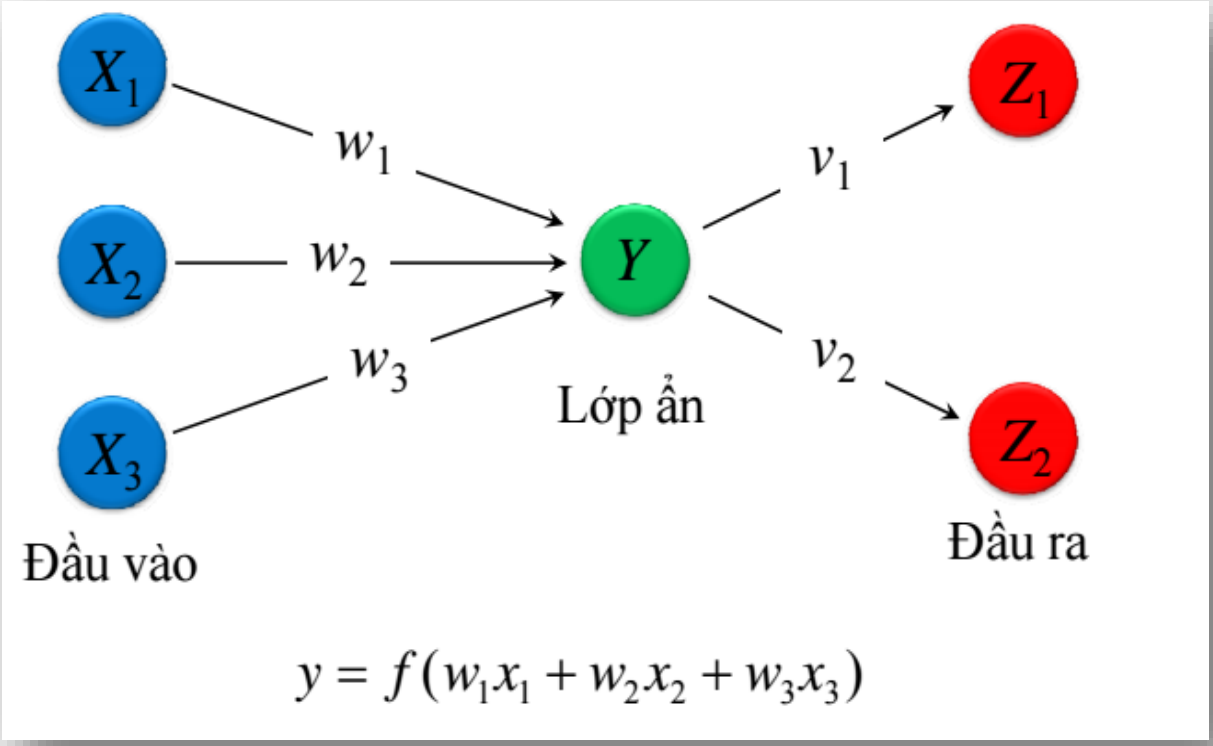
- Chọn vị trí và độ rộng của filter dựa trên trade-off sau:
  - Độ rộng của filter càng lớn thì càng có thêm thông tin về tần số (thấp) nhưng mất thông tin trong miền thời gian (Hình STFT(b)).
  - Độ rộng của filter càng nhỏ thì mất thông tin về tần số (thấp) nhưng có thêm thông tin trong miền thời gian (Hình STFT(c)).

b. Giới thiệu về mạng nơron nhân tạo.

Khái niệm :

Mạng Neuron nhân tạo là mô hình xử lý thông tin được mô phỏng dựa trên hoạt động của hệ thống thần kinh sinh vật, bao gồm số lượng lớn các Neuron được gắn kết để xử lý thông tin. Mạng nơron giống như bộ não con người, được học bởi kinh nghiệm (thông qua huấn luyện), có khả năng lưu giữ những kinh nghiệm hiểu biết (tri thức) và sử dụng những tri thức đó trong việc dự đoán các dữ liệu chưa biết (unseen data) [6].

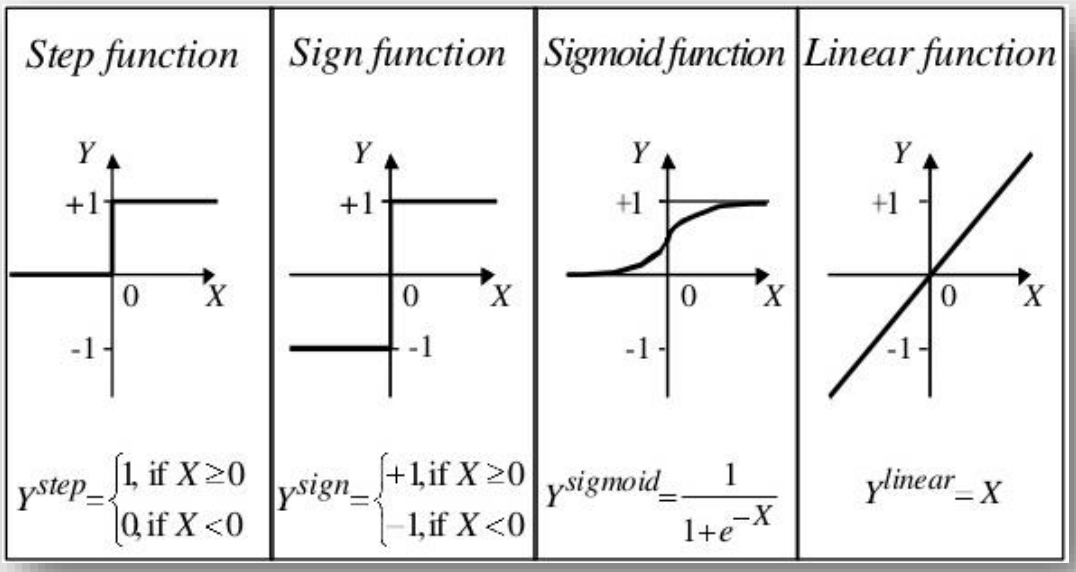
Kiến trúc tổng quát:



***Hình 1**: Kiến trúc mạng nơron nhân tạo [5]*

- **Input:** Các tham số đầu vào
- **Output:** Kết quả đầu ra
- **Trọng số liên kết w1,w2,w3,...:** thể hiện độ mạnh của dữ liệu dùng điều chỉnh tham số đầu vào mỗi nơron
- **Hàm tổng (hàm kết hợp tuyến tính):** y = f(x) tính tổng trọng số của noron, cho biết độ mạnh, khả năng kích hoạt của nơron đó
- **Hàm chuyển đổi (hay hàm kích hoạt):** cho phép xác định output của 1 nơron có được tham gia vào layer tiếp theo không. Một số hàm chuyển đổi: sigmoid, step (threshold), linear,...
- **Lớp ẩn:** 1 mạng nơ ron có thể có 1, 2, 3,... nhiều lớp, thông qua các lớp ẩn, ta có thể thu được kết quả output chính xác hơn.

Dạng hàm kích hoạt:



Hình 2: *Các dạng hàm kích hoạt [5]*

c. Giới thiệu tập dữ liệu âm thanh UrbanSound8k.

Link download tập dữ liệu UrbanSound8k [9]:

https://serv.cusp.nyu.edu/projects/urbansounddataset/index.html

- Là tập dữ liệu dùng phổ biến trong phân loại âm thanh.
- Gồm 10 lớp: air conditioner, car horn, children playing, dog bark, drilling, engine idling, gunshot, jackhammer, siren, và street music.
- Mỗi file là 1 đoạn âm thanh ngắn khoảng 4 giây dùng định dạng *.wav.

2. Nội dung chi tiết:

a. Perceptron

Giới thiệu:

Perceptron là một thuật toán Classification cho trường hợp đơn giản nhất: chỉ có hai class (lớp) (*bài toán với chỉ hai class được gọi là binary classification*) và cũng chỉ hoạt động được trong một trường hợp rất cụ thể. Tuy nhiên, nó là nền tảng cho một mảng lớn quan trọng của Machine Learning là Neural Networks và sau này là Deep Learning [6].

Bài toán:

Bài toán Perceptron được phát biểu như sau: Cho hai class được gán nhãn, hãy tìm một đường phẳng sao cho toàn bộ các điểm thuộc class 1 nằm về 1 phía, toàn bộ các điểm thuộc class 2 nằm về phía còn lại của đường phẳng đó. Với giả định rằng tồn tại một đường thẳng như thế.

Nếu tồn tại một đường phẳng phân chia hai class thì ta gọi hai class đó là linearly separable.

Thuật toán học Perceptron [6]:

Cho tập huấn luyện $D = \{(x_1, d_1), (x_2, d_2), \dots, (x_n, d_n)\}$ với xi là vector thể hiện đối tượng i, d_i là giá trị thể hiện nhóm tương ứng của x_i. Sau khi có tập huấn luyện, ta bắt đầu khởi tạo giá trị cho vector w để tạo nên một hàm f: $f(x) = w^T x = w_1 x_1 + w_n x_n + \cdots + w_n x_n$. Cách thức khởi tạo có thể là ngẫu nhiên hoặc ràng buộc tùy vào người cài đặt. Tuy rằng nếu cách thức khởi tạo tương đối tốt thì quá trình học sẽ nhanh hơn nhưng việc tìm ra một cách thức khởi tạo tốt là khá mơ hồ nên đa phần chúng ta sẽ khởi tạo ngẫu nhiên.

Sau khi đã khởi tạo hàm và có tập huấn luyện, chúng ta sẽ bắt đầu huấn luyện cho máy tính học. Tính các giá trị $f(x)$ với mỗi x_i trong tập huấn luyện (i chạy từ 1 đến n). Kết quả y_i phân nhóm dựa vào hàm f: $f(x)$ sẽ được đem so sánh với giá trị thực d_i. Điều chắc chắn là sẽ có sự sai biệt giữa i và d vì hàm $f(x)$ khởi tạo ban đầu chưa phải là kết quả mong muốn. Để có được hàm $f(x)$ đạt yêu cầu, tiến hàng cập nhật giá trị cho vector w như sau: chọn ra một trong các vector x có y sai khác d và cập nhật $w = w + (d_i - y_i) * x_i$. Bước cập nhật này chính là bước học từ tập huấn luyện.

Sau khi có vector w mới, chúng ta lại tiến hành lặp lại các bước tính toán và cập nhật cho đến khi trung bình các độ sai lệch nhỏ hơn một ngưỡng nào đó cho trước (threshold). Trực quan hóa lên, chúng ta có thể hiểu rằng đường thẳng phân cắt mặt phẳng sẽ được điều chỉnh lên xuống sao cho phân chia hai nhóm riêng biệt một cách chính xác nhất. Chúng ta chỉ có thể tạo ra một hàm f sao cho gần nhất với đáp án thật sự (nghĩa là rất ít các điểm sai lệch) vì việc tạo ra một hàm f thật sự hoàn hảo tương đối khó khăn. Nếu tập huấn luyện có càng nhiều mẫu để huấn luyện với sự phân phối đều (có đối tượng thuộc cả hai nhóm tương đối đều) thì hàm f thu được sẽ càng tốt.

Một lưu ý rằng trong mô tả thuật toán trên, ngần định là hàm f sẽ đi qua gốc tọa độ. Điều này đôi khi sẽ không đúng cho nhiều bài toán và có thể điều chỉnh bằng cách thêm hằng số vào biểu thức: $f: f(x) = c + w * x$(trong đó c là hằng số thêm vào, trường hợp c = 0 sẽ giống như mô tả ban đầu). Các bước còn lại đều không thay đổi.

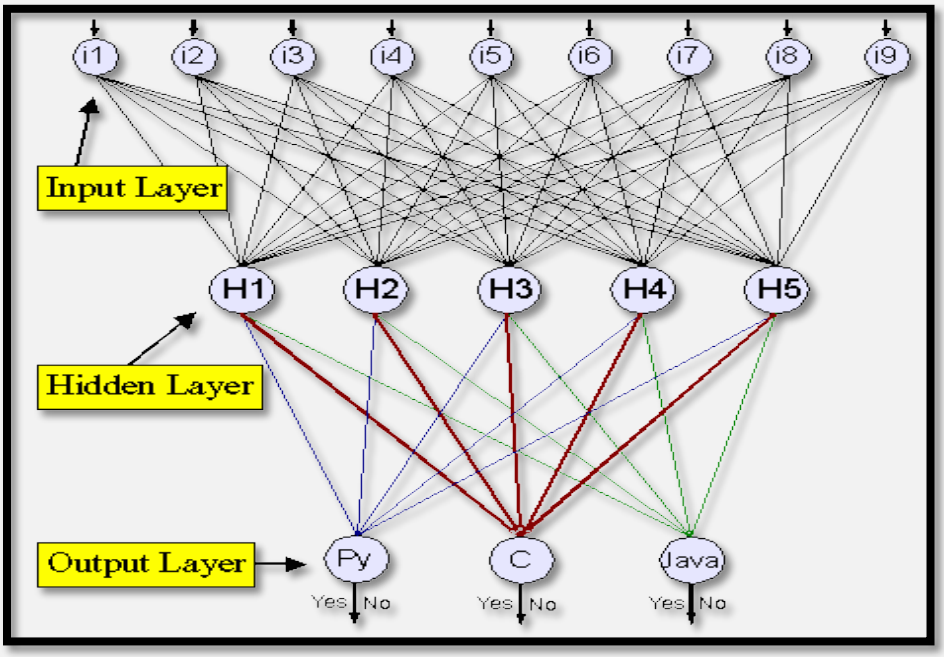
b. Mạng nơron truyền thẳng đa lớp

Giới thiệu:

Là mạng nơron có từ hai lớp ẩn trở lên. Cụ thể: 1 lớp nhập, ít nhất 1 lớp giữa, 1 lớp xuất.

- **Lớp nhập**: nhận tín hiệu đầu vào và tái phân phối đến tất cả các nơ ron trong lớp ẩn [6].
- **Lớp giữa (các lớp ẩn)**: giúp xác định, rút trích các đặc trưng, các nơron biểu diễn các đặc trưng ẩn [6].
- **Lớp xuất**: nhận tín hiệu xuất (các đặc trưng) từ lớp ẩn và xác định mẫu đầu ra [6].

Các nơ ron trong lớp ẩn không thể được quan sát dựa trên đầu vào/ đầu ra của mạng. Vì thế không có cách nào biết rõ đầu ra mong muốn của lớp ẩn là gì.



Hình 3: Mạng nơron truyền thẳng đa lớp [5]

Phương pháp học

Mạng neural nhân tạo phỏng theo việc xử lý thông tin của bộ não người, do vậy đặc trưng cơ bản của mạng là có khả năng học, khả năng tái tạo các hình ảnh và dữ liệu khi đã học. Trong trạng thái học thông tin được lan truyền theo hai chiều nhiều lần để học các trọng số. Có 3 kiểu học chính, mỗi kiểu học tương ứng với một nhiệm vụ học trừu tượng. Đó là học có giám sát (có mẫu), học không giám sát và học tăng cường. Thông thường loại kiến trúc mạng nào cũng có thể dùng được cho các nhiệm vụ.

Học có giám sát.

Một thành phần không thể thiếu của phương pháp này là sự có mặt của một người thầy (ở bên ngoài hệ thống). Người thầy này có kiến thức về môi trường thể hiện qua một tập hợp các cặp đầu vào - đầu ra đã được biết trước. Hệ thống học (ở đây là mạng neural) sẽ phải tìm cách thay đổi các tham số bên trong của mình (các trọng số và các ngưỡng) để tạo nên một ánh xạ có khả năng ánh xạ các đầu vào thành các đầu ra mong muốn. Sự thay đổi này được tiến hành nhờ việc so sánh giữa đầu ra thực sự và đầu ra mong muốn.

Học không giám sát.

Trong học không có giám sát, ta được cho trước một số dữ liệu x và hàm chi phí cần được cực tiểu hóa có thể là một hàm bất kỳ của dữ liệu x và đầu ra của mạng, f – hàm chi phí được quyết định bởi phát biểu của bài toán. Phần lớn các ứng dụng nằm trong vùng của các bài toán ước lượng như mô hình hóa thống kê, nén, lọc, phân cụm.

Học tăng cường.

Dữ liệu x thường không được tạo trước mà được tạo ra trong quá trình một agent tương tác với môi trường. Tại mỗi thời điểm t, agent thực hiện hành động yt và môi trường tạo một quan sát xt với một chi phí tức thời Ct, theo một quy trình động nào đó (thường là không được biết). Mục tiêu là một sách lược lựa chọn hành động để cực tiểu hóa một chi phí dài hạn nào đó, nghĩa là chi phí tích lũy mong đợi. Quy trình hoạt động của môi trường và chi phí dài hạn cho mỗi sách lược thường không được biết, nhưng có thể ước lượng được. Mạng neural nhân tạo thường được dùng trong học tăng cường như một phần của thuật toán toàn cục. Các bài toán thường được giải quyết bằng học tăng cường là các bài toán điều khiển, trò chơi và các nhiệm vụ quyết định tuần tự (sequential decision making) khác.

Thuật toán huấn luyện mạng – thuật toán lan truyền ngược [5]:

- **Bước 1**: Tạo mạng truyền thẳng có n_{in} Nơron đầu vào, n_{hiden} Nơron trên mỗi lớp ẩn và h lớp ẩn trong mạng, với n_{out} đầu ra
- **Bước 2**: Khởi tạo bộ trọng cho mạng với giá trị nhỏ
- **Bước 3**: Trong khi <điều kiện kết thúc chưa thoả> làm:

  Với mỗi cặp (x, t) trong không gian mẫu huấn luyện thực hiện:

- Xét lớp nhập (*): truyền x qua mạng, tại mỗi lớp xác định đầu ra của mỗi Nơron. Quá trình này được thực hiện đến lớp xuất dựa theo cấu trúc mạng cụ thể.
- Xét lớp xuất: đối với đầu ra o_k của Nơron k trong lớp xuất K, xác định sai số: $\sigma_k = o_k(1-o_k)(t_k-o_k)$
- Chuyển sang lớp ẩn L kế nó, đặt L=K-1
- Xét các lớp ẩn (**): với mỗi Nơron l trên lớp ẩn thứ L, xác định sai số:

$$\sigma_l = o_l(1-o_l)\sum_{i\in L+1} w_{il}\sigma_i$$

- Cập nhật lại trọng số có trong mạng, w_{ji}
  - $w_{ji} = w_{ji} + \Delta w_{ji}$ với $\Delta w_{ji} = \eta\sigma_j o_{ji}$
- *Nếu (L>1) thì: chuyển sang lớp ẩn trên nó : L=L-1 và quay lại bước (**)*
- *Ngược lại: chọn cặp (x, t) mới trong không gian mẫu học, quay lại bước (*).*

c. Mạng nơron tích chập – CNN

Mạng nơ ron tích chập được ứng dụng nhiều trong xử lý ảnh và xử lý tín hiệu.

Đặc trưng- Feature:

CNNs so sánh hình ảnh theo từng mảnh. Các mảnh mà nó tìm được gọi là các feature. Bằng cách tìm ở mức thô các feature khớp nhau ở cùng vị trí trong hai hình ảnh, CNNs nhìn ra sự tương đồng tốt hơn nhiều so với việc khớp toàn bộ bức ảnh.

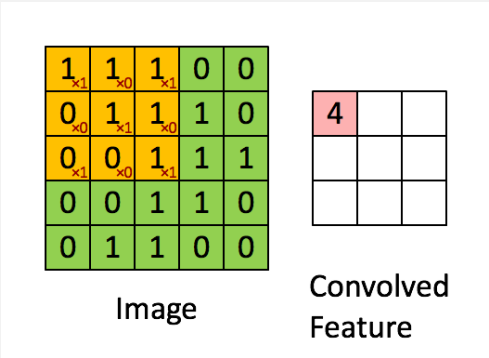
Mỗi feature giống như một hình ảnh mini - một mảng hai chiều nhỏ. Các feature khớp với các khía cạnh chung của các bức ảnh

Tích chập – Convolution [7]:

Convolution là một phép toán quan trọng biểu diễn cho các hệ thống rời rạc tuyến tính. Tín hiệu sau khi qua hệ thống xử lý bằng các phép convolution với ma trận đáp ứng xung (impulse response) nào đó sẽ cho ra kết quả là tín hiệu đầu ra. Xét trên khía cạnh thực hành, thiết kế một hệ thống H là xác định cách tính Convolution và nguyên tắc mà hệ thống đối xử với tín hiệu (ma trận đáp ứng). Việc triển khai một cách chính xác phép tính này sẽ trợ giúp đắc lực trong quá trình xử tín hiệu, ảnh, ...

Để dễ hình dung, ta áp dụng 1 kernel tích chập 1 bước ảnh trắng đen có giá trị mỗi pixel là 0 hoặc 1. Thực hiện theo quy tắc element-wise (nhân từng thành phần) và sau đó lấy tổng các thành phần đó. Kết quả ra được là 1 ma trận có kích thước bằng kích thước ảnh đem tích chập.

Ví dụ:



Hình 4: Minh họa bài toán tích chập

Với ma trận ảnh tích chập như trên và kernel là:

$$\begin{matrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{matrix}$$

Gọi thứ tự các giá trị trên ảnh từ trái sang, từ trên xuống là $x_0, x_2, \ldots x_8$

Gọi thứ tự các giá trị trên kernel từ trái sang, từ trên xuống là $y_0, y_2, \ldots y_8$

Thực hiện tích chập theo quy tắc element-wise:

Ta có: value = $x_0*y_8 + x_1*y_7 + \ldots + x_8*y_0$ = 1*1 + 1*0 + 1*1 + 0*0 + 1*1 + 1*0 + 0* 1 + 0*0 + 1* 1 = 4

Khi xem một hình ảnh mới, CNN không biết chính xác nơi các feature này sẽ khớp nên nó sẽ thử chúng khắp mọi nơi, ở mọi vị trí có thể. Khi tính toán sự khớp của một feature trên toàn bộ ảnh, chúng ta làm thành một filter kernel (nhân bộ lọc). Phần toán ta sử dụng để làm điều này được gọi là tích chập, từ đó mà Mạng Nơ-ron Tích chập (Convolutional Neural Networks) có tên là như vậy.

Pooling [1]:

Pooling là một cách lấy những hình ảnh lớn và làm co chúng lại trong khi vẫn giữ các thông tin quan trọng nhất trong đó. Nó bao gồm việc duyệt bước một ô vuông cửa sổ nhỏ dọc trên một hình ảnh và lấy giá trị lớn nhất từ cửa sổ ở mỗi bước. Trong thực tế, một cửa sổ có cạnh 2 hoặc 3 điểm ảnh và duyệt bước mỗi 2 điểm ảnh là được.

Sau khi pooling, một hình ảnh sẽ có khoảng một phần tư số điểm ảnh so với lúc bắt đầu. Vì nó giữ các giá trị lớn nhất từ mỗi cửa sổ, nó sẽ bảo toàn tính khớp của mỗi feature bên trong cửa sổ. Nghĩa là nó không quan tâm quá nhiều về vị trí chính xác nơi feature khớp, miễn là nó khớp ở chỗ nào đó trong cửa sổ. Kết quả là CNNs có thể tìm xem liệu một feature có nằm trong hình ảnh mà không cần lo nó nằm ở đâu.

Rectified Linear Units – ReLU [1]

ReLU hoạt động rất đơn giản cứ nơi nào có số âm, hoán đổi nó với 0. Điều này giúp CNN giữ vững sự tin cậy toán học bằng cách giữ các giá trị đã được học khỏi bị mắc kẹt gần 0 hoặc bị thổi bay về vô tận. Tuy ReLU đơn giản nhưng nếu không có nó, chúng sẽ không đi xa hơn được.

Sau khi thực hiện Tích chập – Pooling và ReLU với một số lần lập nhất định ta được một mạng liên kết đầy đủ Fully Connected:

Đến bước này, ta có thể áp dụng các tính chất và các thuật toán xử lý của mạng nơron truyền thẳng để tiến hành giải quyết bài toán theo yêu cầu.

Các siêu tham số:

Tương tự như các mô hình máy học khác, siêu tham số vẫn là vấn đề quan tâm hàng đầu ảnh hưởng trực tiếp đến kết quả học.

- Đối với mỗi layer tích chập, bao nhiêu đặc trưng (nodes) là đủ?
- Đối với mỗi layer pooling, kích cỡ ô vuông cửa sổ như thế nào? duyệt mỗi bước bao nhiêu?
- Đối với mỗi các layer được kết nối đầy đủ thêm vào, bao nhiêu nơron ẩn?
- Ngoài ra còn có những quyết định về kiến trúc ở cấp cao hơn cần thực hiện: Số lượng mỗi layer thêm vào là bao nhiêu? Theo thứ tự nào?
- …

d. Áp dụng mạng nơron tích chập trong xử lý âm thanh

Mượn ý tưởng về ma trận các pixel trong ảnh, thực hiện tiền xử lý với đơn vị nhỏ nhất là điểm phổ đồ 1hz/1ms, từ đó xây dựng các ma trận phổ đồ rời rạc.

Giai đoạn tiền xử lý [1]:

Tiền xử lý Log Power spectrogram của các đoạn âm thanh:

- Mỗi spectrogram chia ra 60x41 (bands x frames) data frames. Mỗi data frames gọi là một ma trận phổ đồ rời rạc.
- Tương tự kênh R,B,G trong ảnh, âm thanh cũng có 2 kênh là spect và thelta.

Bản chất phổ đồ là 1 ảnh cho biết giá trị tần số, thời gian và cường độ, ta rời rạc hóa phổ đồ tức là chia phổ đồ thành 1 ma trận gồm m x n các ô, mỗi ô sẽ cho biết tần số tín hiệu, khoản thời gian tín hiệu truyền được trong ô và độ lớn âm thanh.

Mỗi data frames có thể hiểu đơn giản là một ảnh với thông tin theo trục Ox là frames đại diện cho 1 khung thời gian nhất định và theo trục Oy là bands đại diện cho 1 dãy tần số nhất định. Data frames sẽ thể hiện màu đặc trưng cho độ lớn của âm thanh (cường độ âm).

Spect và thelta là hai kênh tác động lênh cường độ âm tương tự R,B,G là 3 kênh tác động lên màu ảnh.

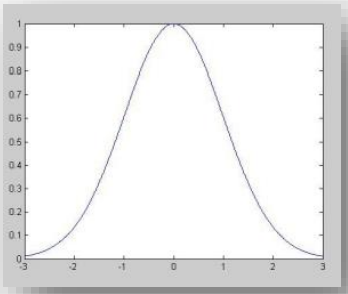
60x41 là giá trị chọn để chia nhỏ spectrogram được thử nghiệm nhiều và cho kết quả tốt.

Giai đoạn thực thi:

Lớp Convolution [1]:

Bước 1: Khởi đầu:

Tích chập Gaussian filter (3x3) với ma trận phổ đồ rời rạc của các đoạn âm thanh đã qua tiền xử lý. Tích chập lưu ý zero padding. Ma trận phổ đồ được tách làm 2, ứng với 2 kênh spec và thelta.

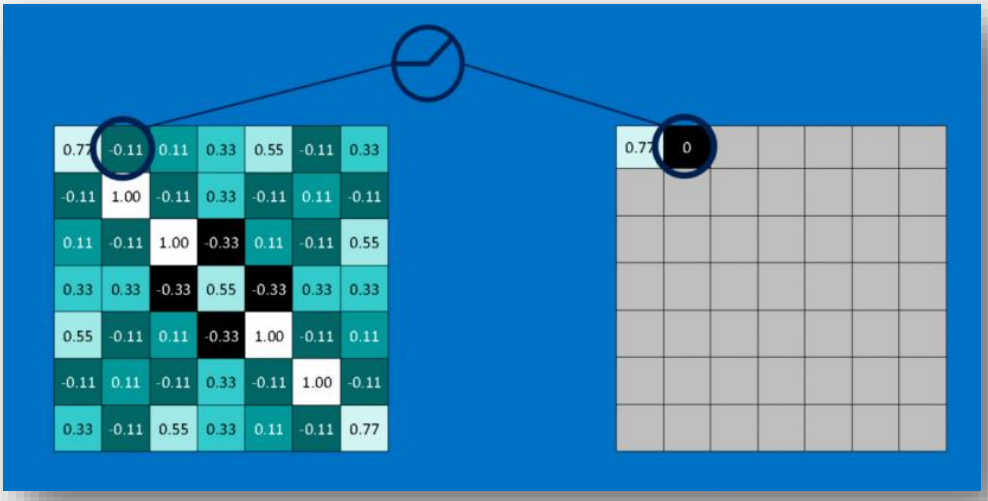


1	2	1
2	4	2
1	2	1

* 1 / 16

Hình 5: Công thức hàm Gauss: $g_\sigma(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\left(-\frac{x^2}{2\sigma^2}\right)}$ và bộ lọc Gauss (Gaussian Filter) sau khi rời rạc hóa thành ma trận 3x3.

Bước 2: ReLU (Rectifiled Linear Unit): dùng hàm kích hoạt f(x) = max(0,x) tạo tính chất phi tuyến biến các giá trị âm sau kích hoạt thành giá trị 0.



Hình 6: Minh họa ReLU[3]

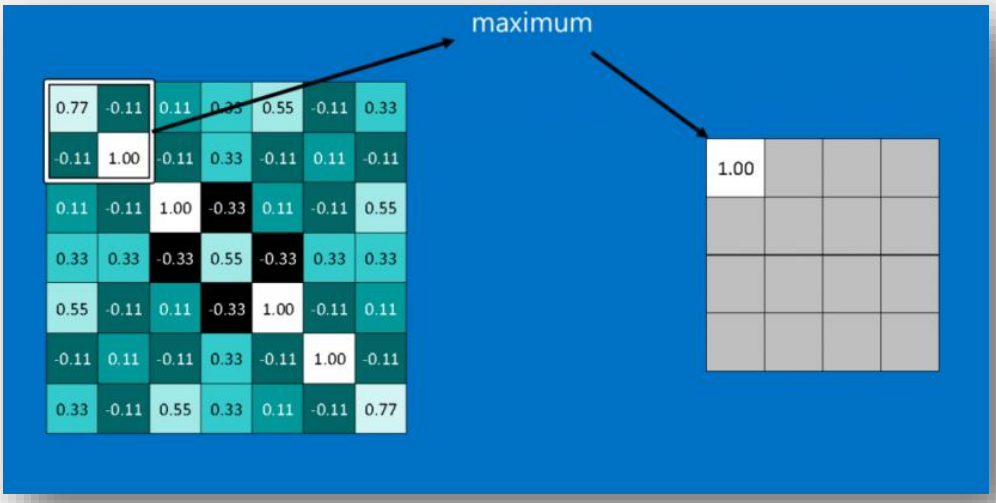
Bước 3: Pooling giảm chiều:

Trượt 1 cửa sổ có kích thước m*n lên các ma trận phổ đồ đã thực hiện tích chập. Thực hiện lấy giá trị đại diện (giá trị lớn nhất - maxpooling) trong vùng trượt đến.

Ưu điểm pooling:

- Hạn chế overfitting.
- Cải thiện tốc độ xử lý.
- Giữ lại đặc trưng bất biến.

Ở đây, ta chọn cửa sổ trượt là 4x2 hoặc 2x2, đây kích thước của cửa sổ trượt được kiểm tra thực nghiệm và thấy có hiệu quả.



Hình 7: Minh họa Pooling giảm chiều [3]

Bước 4: lặp lại bước 1 đến 3 với số lần nhất định.

- ❑ Thực hiện lặp lại quá trình:

  Convolution → ReLU → Pooling giảm chiều.

- ❑ Mục đích: nâng cao kết quả đầu ra.
- ❑ Lưu ý: Thực hiện lặp quá trình với 1 số lượng vừa đủ vì lập quá nhiều sẽ làm mất đặc trưng khiến xảy ra underfitting.

Sau khi hoàn tất bước pooling. Do tính chất ReLU là phi tuyến vì thế ta xem mỗi kênh đã được tách và xử lý tích chập sau bước này như 1 ảnh phổ đặc trưng. Từ ảnh phổ đặc trưng đó, ta tiếp tục chia ma trận phổ đồ làm hai kênh và thực hiện tích chập. Việc này khiến cho số lượng ảnh phổ xử lý qua mỗi lớp tích chập ngày càng tăng nhưng kích thước lại ngày càng giảm.

Lớp Fully Connected [1]:

Sau khi hoàn tất lớp tích chập cuối cùng, ta được một nhóm các ảnh phổ đặc trưng. Các ảnh phổ này có mối quan hệ chặc chẽ nhau vì đã qua nhiều lớp tích chập (gọi là đặc trưng bất biến cao). Các đặc trưng này được xem như đầu vào cho mạng nơron, từ đó ta áp dụng thuật giải lan truyền ngược để xử lý cập nhật trọng số theo các nhãn tương tự như việc học có giám sát trên mạng nơron truyền thẳng.

Các siêu tham số [1]:

Tổng hợp lại các siêu tham số áp dụng vào bài toán phân tích âm thanh dùng mạng nơron tích chập.

- Có tổng cộng 5 layer, bao gồm: 3 layer tích chập, 1 layer liên kết đầy đủ, 1 layer phân loại.
- Layer 1 có 24 nodes, layer 2 và 3 có 48 nodes, layer 4 có 64 nodes, layer 5 có 10 nodes ứng với số phân lớp.
- Spectrogram được chia ra làm 60 x 41 data frames.
- Kernel tích chập: dùng Gaussian filter 3x3.
- Pooling giảm chiều sử dụng cửa sổ trượt 4x2 hoặc 2x2.

Các siêu tham số trên đã được thực nghiệm và cho kết quả khá tốt khi áp dụng vào bài toán.

IV. Thực nghiệm

Áp dụng mô hình mạng nơ ron tích chập theo hướng xử lý 5 lớp mạng [1] bao gồm: 3 lớp tích chập, 1 lớp liên kết đầy đủ và 1 lớp phân loại. Áp dụng trên tập dữ liệu UrbanSound8k với 8732 mẫu chia cho 3 tập dữ liệu là tập train, validation và test.

Kết quả: độ chính xác đạt được là 78%

```
Train on 43209 samples, validate on 5314 samples
Epoch 1/50
43209/43209 [==============================] - 110s - loss: 1.5326 - acc: 0.4630 - val_loss: 0.9697 - val_acc: 0.7439
Epoch 2/50
43209/43209 [==============================] - 113s - loss: 1.5289 - acc: 0.4675 - val_loss: 0.9373 - val_acc: 0.7614
Epoch 3/50
43209/43209 [==============================] - 112s - loss: 1.5259 - acc: 0.4652 - val_loss: 0.9345 - val_acc: 0.7742
Epoch 4/50
43209/43209 [==============================] - 111s - loss: 1.5105 - acc: 0.4690 - val_loss: 0.8699 - val_acc: 0.7855
Epoch 5/50
43209/43209 [==============================] - 106s - loss: 1.5035 - acc: 0.4706 - val_loss: 1.0899 - val_acc: 0.7036
Epoch 6/50
43209/43209 [==============================] - 108s - loss: 1.4940 - acc: 0.4741 - val_loss: 0.8507 - val_acc: 0.7977
Epoch 7/50
43209/43209 [==============================] - 111s - loss: 1.4985 - acc: 0.4720 - val_loss: 0.8457 - val_acc: 0.7919
Epoch 8/50
43209/43209 [==============================] - 110s - loss: 1.4922 - acc: 0.4717 - val_loss: 0.8830 - val_acc: 0.7768
Epoch 9/50
43209/43209 [==============================] - 109s - loss: 1.4859 - acc: 0.4726 - val_loss: 0.8295 - val_acc: 0.7966
Epoch 10/50
43209/43209 [==============================] - 105s - loss: 1.4788 - acc: 0.4759 - val_loss: 0.8189 - val_acc: 0.7983
Epoch 11/50
43209/43209 [==============================] - 108s - loss: 1.4786 - acc: 0.4750 - val_loss: 0.8420 - val_acc: 0.7930
Epoch 12/50
43209/43209 [==============================] - 107s - loss: 1.4664 - acc: 0.4818 - val_loss: 0.8674 - val_acc: 0.7858
```

Hình 8: Quá trình huấn luyện.

Hình 9: Kết quả kiểm tra trên tập test

Nhận xét:Bằng việc sử dụng mô hình mà tác giả Karol.J.Piczak đề ra [2] và cải tiến tăng thêm số lượng lớp tích chập theo mô hình của tác giả Justin Salamon và Juan Pablo Bello, ta đã đạt được kết quả khá khả quan.

Một số dữ liệu so sánh:

- Tác giả Karol.J.Piczak áp dụng mô hình mình đề ra và đạt được độ chính xác là 73.6%[2].
- Tác giả Shuhui Qu, Juncheng Li, Wei Dai, Samarjit Das áp dụng mô hình mạng nơron tích chập kết hợp pipeline đạt độ chính xác lên đến 78.34%[3].
- Tác giả Justin Salamon và Juan Pablo Bello đã thực nghiệm trên mô hình SKM, PiczakCNN và SB-CNN và đạt được độ chính xác lần lượt là: 0.74, 0.73 và 0.73[1].
- Tác giả Justin Salamon áp dụng mô hình mình đề ra và đạt được độ chính xác lên đến 84%[1].

V. Kết luận

- Nhận dạng âm thanh là một chủ đề có nhiều tiềm năng ứng dụng và thực tiễn hỗ trợ cho nhiều loại hệ thống nhằm mục đích phục vụ các nhu cầu khác nhau của con người trên nhiều lĩnh vực bên cạnh lĩnh vực công nghệ thông tin như: an ninh, quốc phòng, y tế, giáo dục, giám sát, kinh tế - xã hội, thương nghiệp và kinh doanh,…
- Thông qua việc tìm hiểu các kĩ thuật xử lý âm cũng như kĩ thuật học bằng mạng nơron tích chập, ta đã đạt được một lượng lớn các kiến thức mà các kiến thức này có thể được sử dụng trong nhiều lĩnh vực khác nhau, từ đó tạo tiền đề và nền tảng để tìm hiểu những kĩ thuật xử lý cao hơn ở các lĩnh vực phức tạp hơn.
- Mặc dù vẫn gặp phải những khó khăn nhất định, tuy nhiên ta vẫn hoàn tất nội dung tìm hiểu và thực nghiệm mô hình cho được một kết quả tương đối tốt.

VI. Tài liệu tham khảo

TÀI LIỆU THAM KHẢO:

[1] Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification - Justin Salamon, Juan Pablo Bello.

[2] Evironmental sound classification with convolution neural networks – Karol J.Piczak.

[3] Understanding Audio Pattern Using Convolution Neural Network from Raw Waveform - Shuhui Qu, Juncheng Li, Wei Dai, Samarjit Das.

[4] Audio Event Classification Using Deep Neural Networks - Minkyu Lim, Donghyun Lee, Hosung Park, Unsang Park and Ji-Hwan Kim, Jeong-Sik Park , Gil-Jin Jang.

[5] Slide bài giảng tuần 4 về mạng nơ ron nhân tạo - Tiến sĩ Lê Hoàng Thái.

[6] Giáo trình Cơ sở trí tuệ nhân tạo – Phó giáo sư, Tiến sĩ Lê Hoài Bắc, Thạc sĩ Tô Hoài Việt.

LINK THAM KHẢO ONLINE:

[7] Recognizing Sounds (A Deep Learning Case Study) – Arthur Juliani:

https://medium.com/@awjuliani/recognizing-sounds-a-deep-learning-case-study-1bc37444d44d#.k0pckvg6a

[8] Convolution Neural Network 1 and 2:

https://viblo.asia/

[9] UrbanSound8k:

https://serv.cusp.nyu.edu/projects/urbansounddataset/urbansound8k.html

Lưu ý: *Các nội dung được đánh chỉ mục là nội dung được tham khảo trực tiếp đến (có thể được dịch lại, được sửa chữa, bổ sung hoặc tóm tắt lại cho phù hợp với bài toán đang được xử lý)*