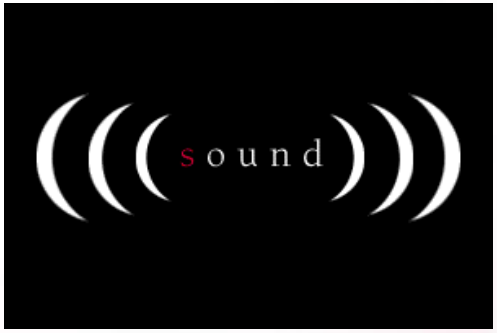
NHẬN DẠNG ÂM THANH DÙNG NEURAL NETWORK

Thực hiện: Đỗ Khánh Long Tường

Khoa Công Nghệ Thông Tin
Đại học Khoa Học Tự Nhiên

NỘI DUNG TRÌNH BÀY



I. Giới thiệu và khái niệm
   - ❑ Âm thanh
   - ❑ Tập dữ liệu UrbanSound8K
   - ❑ Mạng Neural nhân tạo
II. Mạng Neural truyền thẳng đa lớp
III. Mạng Neural tích chập (CNN)
IV. Ứng dụng
V. Tài liệu tham khảo
VI. Kết quả thực nghiệm
VII. Demo

I. Giới thiệu và khái niệm

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 được đặc trưng bởi **tần số**, **bước sóng**, **chu kỳ**, **biên độ** và **vận tốc lan truyền** (tốc độ âm thanh).
- *Đặc tính của âm thanh:* ***sóng dọc***

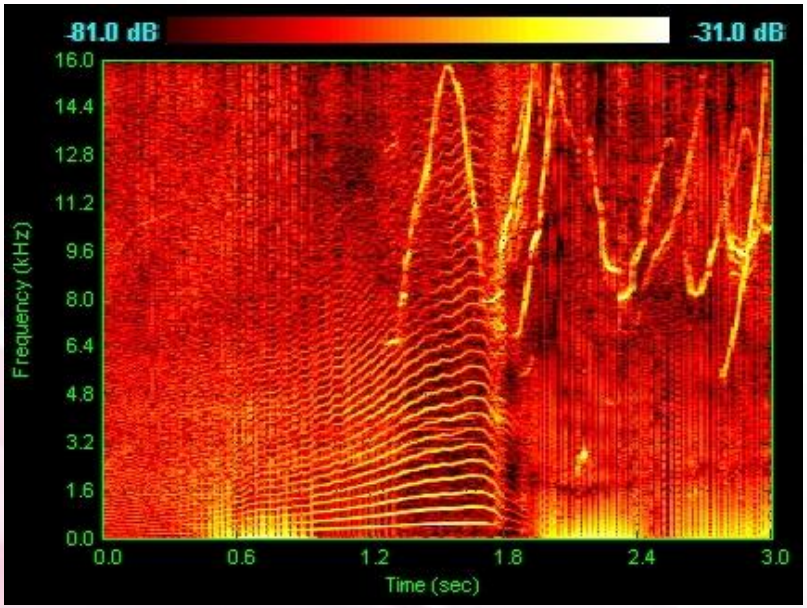
I. Giới thiệu và khái niệm

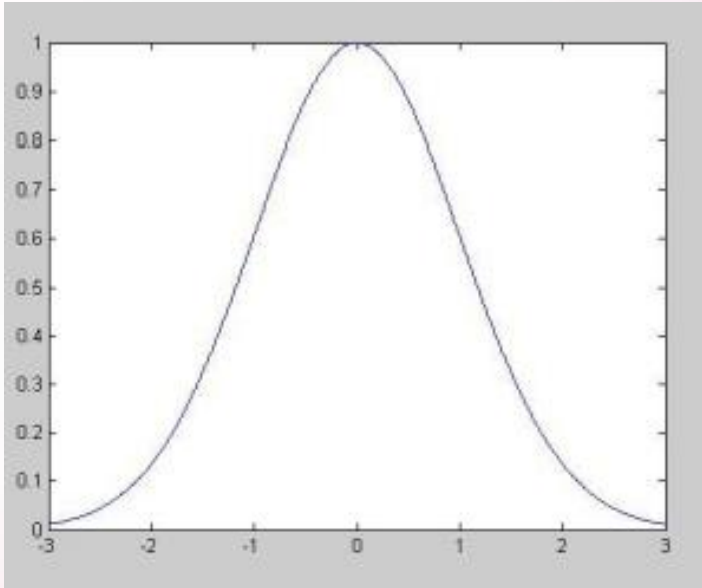
TÍN HIỆU ÂM THANH:

- **Tín hiệu sóng (Waveplot):** cho thông tin về tần số bước sóng của âm thanh.
- **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).
- **Log Power Spectrogram**: là phổ đồ Gaussian Kernel có độ rộng thay đổi theo hàm log.

I. Giới thiệu và khái niệm

TÍN HIỆU ÂM THANH:





Spectrogram

$$g_\sigma(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\left(-\frac{x^2}{2\sigma^2}\right)}$$

I. Giới thiệu và khái niệm

Phép biến đổi Fourier thuận rời rạc

$$F(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-2\pi i \left(\frac{xu}{M} + \frac{yv}{N} \right)}$$

$$u = 0,1,\ldots, M-1 \qquad \text{v} = 0,1,\ldots, N-1$$

Phép biến đổi Fourier ngược rời rạc

$$f(x,y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) e^{2\pi i \left(\frac{xu}{M} + \frac{yv}{N} \right)}$$

$$x = 0,1,\ldots, M-1 \qquad \text{y} = 0,1,\ldots, N-1$$

I. Giới thiệu và khái niệm

Tập dữ liệu UrbanSound8k:

- 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.

I. Giới thiệu và khái niệm

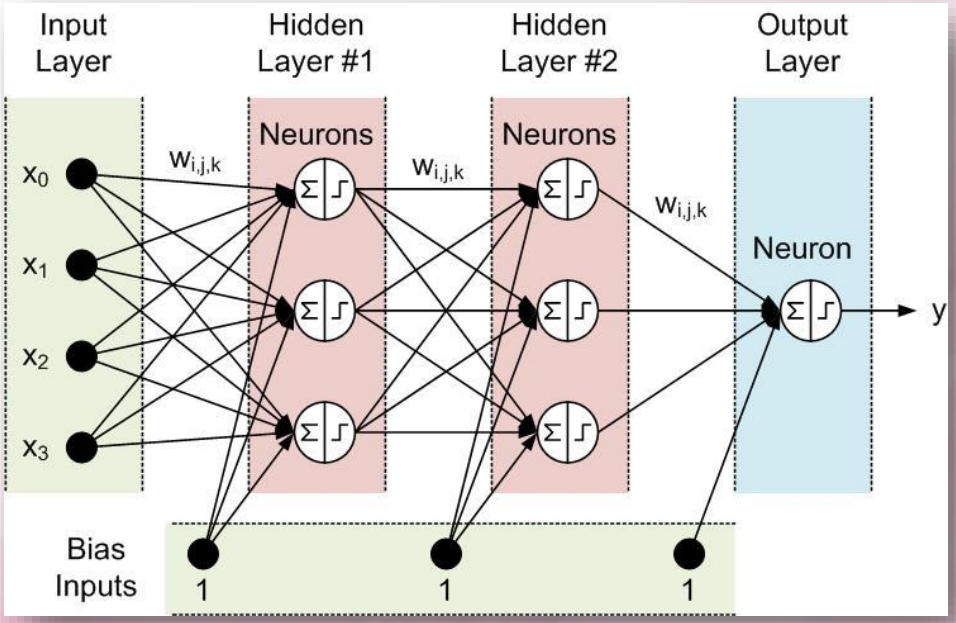
Mạng Neural nhân tạo:

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).

I. Giới thiệu và khái niệm

Kiến trúc tổng quát:

- Input Layer
- Output Layer
- Connection Weight (Trọng số liên kết)
- Summation Funct (Hàm tổng)
- Transformation Funct (Hàm kích hoạt)
- Hidden Layer



I. Giới thiệu và khái niệm

Các dạng huấn luyện phổ biến:

Học có giám sát

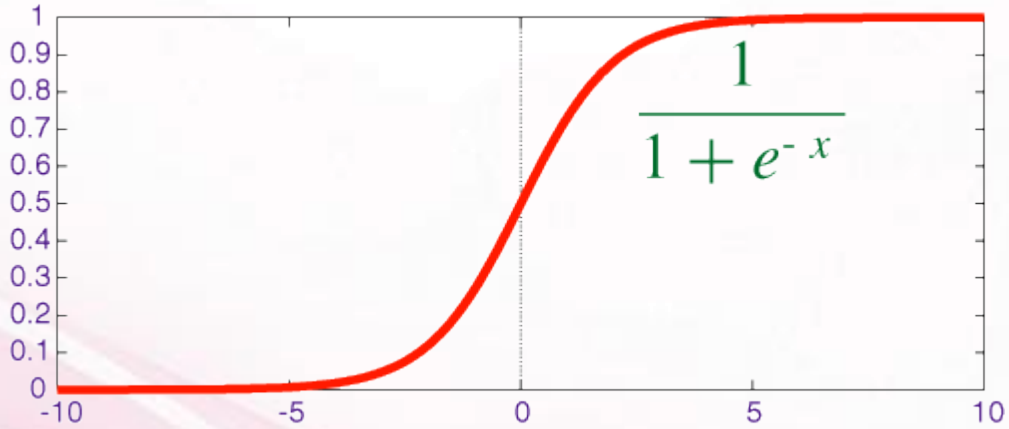
- Véctơ huấn luyện được kèm theo véctơ mục tiêu.
- Các trọng số được điều chỉnh theo một thuật toán huấn luyện.

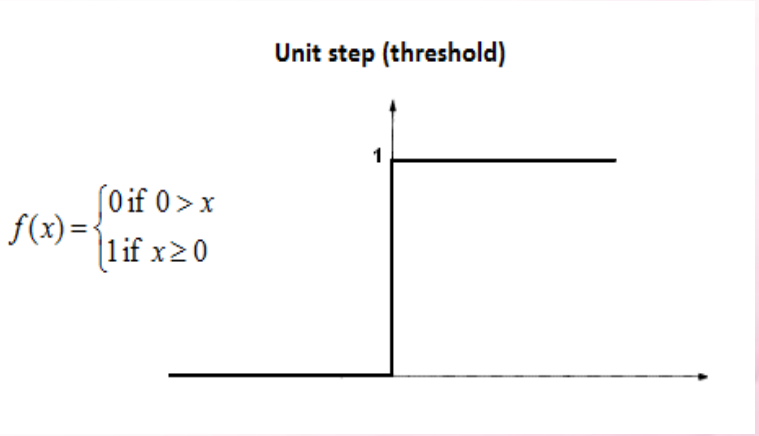
Học không giám sát

- Mạng tự tổ chức nhóm các véctơ đầu vào với nhau, không cần biết đặc điểm của (mỗi) nhóm.
- Không có véctơ mục tiêu.
- Mạng tự tạo ra véctơ đặc trưng của từng nhóm.

I. Giới thiệu và khái niệm

Phân lớp:



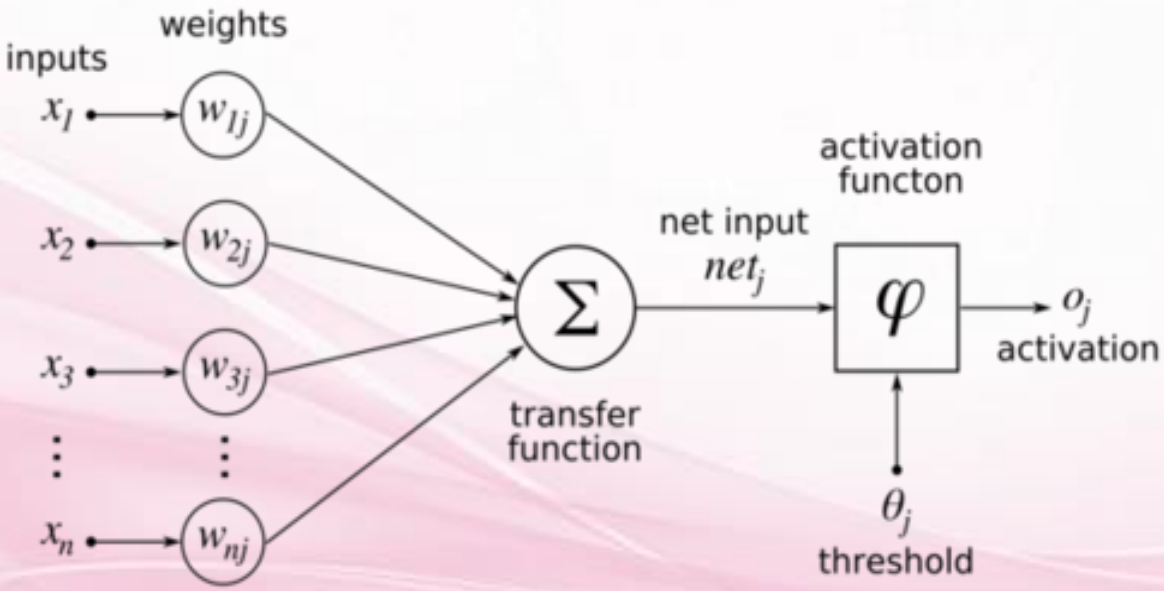


$$Sigmoid: y = \frac{1}{1 + e^{-x}}$$

$$Step: y = \begin{cases} 1 \text{ nếu } x \geq 0 \\ 0 \text{ nếu } x < 0 \end{cases}$$

II. Mạng nơron truyền thẳng đa lớp

Perceptron:



- Dạng đơn giản nhất của 1 nơron nhân tạo.
- Bao gồm 1 nơ-ron duy nhất với các trọng số có thể điều chỉnh được và một ngưỡng cố định (threshold).

Mạng nơron đa lớp:

- 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.
  - **Lớp giữa (các lớp ẩn)**: giúp xác định, rút trích các đặc trưng, các nơron biểu diễn các đặc trưng ẩn.
  - **Lớp xuất**: nhận tín hiệu xuất (các đặc trưng) từ lớp ẩn và xác định mẫu đầu ra.
- 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.

II. Mạng nơron truyền thẳng đa lớp

Deep Network:

- Hay gọi là mạng nơron sâu là một mạng nơron nhân tạo với nhiều đơn vị lớp ẩn giữa lớp nhập và lớp xuất.
- Thực hiện phương pháp xử lý tính toán riêng biệt kết hợp hàm phi tuyến dùng thuật toán lan truyền ngược.
- Một số loại Deep Network thường thấy:
  - Recurent Neural Network (mạng nơ ron tái phát) dùng trong mô hình hóa ngôn ngữ.
  - Convolution Neural Network (mạng nơron tích chập) dùng trong thị giác máy tính và xử lý tín hiệu.

II. Mạng nơron truyền thẳng đa lớp

Đặc điểm Deep Network:

- Các trọng số đầu vào không còn được khởi tạo ngẫu nhiên mà sẽ được khởi tạo dựa trên mô hình học không giám sát.
- Vẫn sử dụng thuật giải lan truyền ngược để tiến hành cập nhật trọng số.
- Các mô hình deep network thường khác nhau ở phương pháp học không giám sát (gom cụm, chia nhóm hay rút trích đặc trưng).

II. Mạng nơron truyền thẳng đa lớp

Thuật giải lan truyền ngược:

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}

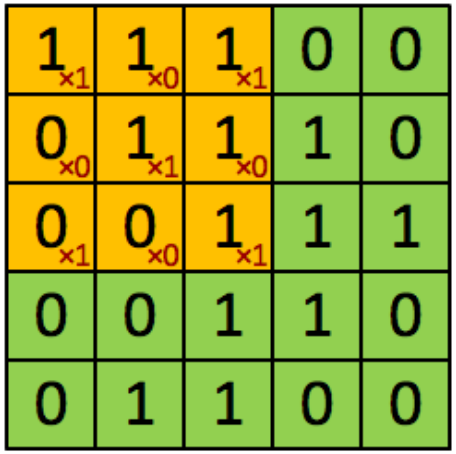
  $w_{ji} = w_{ji} + \Delta w_{ji}$ với $\Delta w_{ji} = \eta \sigma_j o_{ji}$
- ***Nếu (L>1) thì:*** *chuyển sang lớp ẩn trên nó : L=L-1 và quay lại bước* ***(**)***

  Ngược lại: chọn cặp (x, t) mới trong không gian mẫu học, quay lại bước ***(*)***.

III. Mạng nơron tích chập

Tích chập:

- Phương pháp toán thường sử dụng trong xử lý tín hiệu.
- Thực hiện bằng cách nhân các giá trị xét với giá trị tương ứng trong kernel và thực hiện lấy tổng các giá trị tính được.

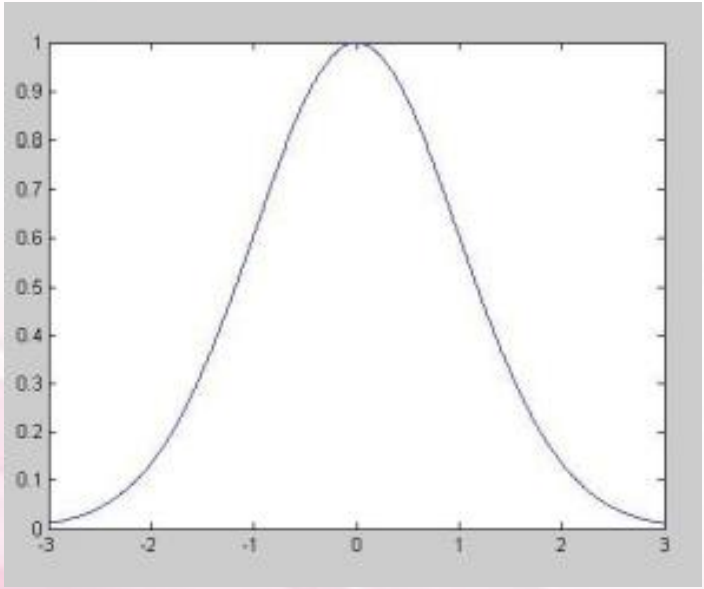


Tích chập trong mạng nơron:

- Các layer liên kết được với nhau thông qua cơ chế convolution.
- Layer tiếp theo là kết quả convolution từ layer trước đó, nhờ vậy mà ta có được các kết nối cục bộ.
- Nghĩa là mỗi nơron ở layer tiếp theo sinh ra từ filter áp đặt lên một vùng cục bộ của nơ-ron layer trước đó.

III. Mạng nơron tích chập

Gaussian Filter:



1	2	1
2	4	2
1	2	1

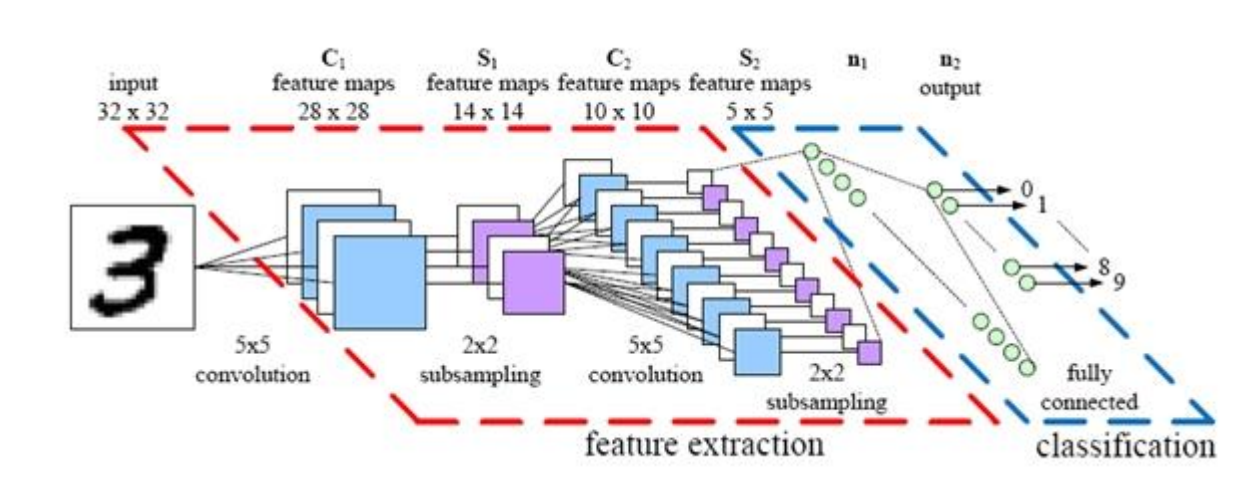
* 1 / 16

$$g_\sigma(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\left(-\frac{x^2}{2\sigma^2}\right)}$$

Gaussian Filter
(Rời rạc hóa thành
ma trận Gaussian 3x3)

Ưu điểm mạng nơron tích chập

- Lọc nhiễu tốt.
- Đặc trưng thu được có tính bất biến cao.
- Tính kết hợp cục bộ tốt.



Áp dụng vào 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:**

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.

Áp dụng vào âm thanh

Giai đoạn thực thi:

Lớp Convolution:

- **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.
- **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.

Áp dụng vào âm thanh

- **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.

Áp dụng vào âm thanh

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

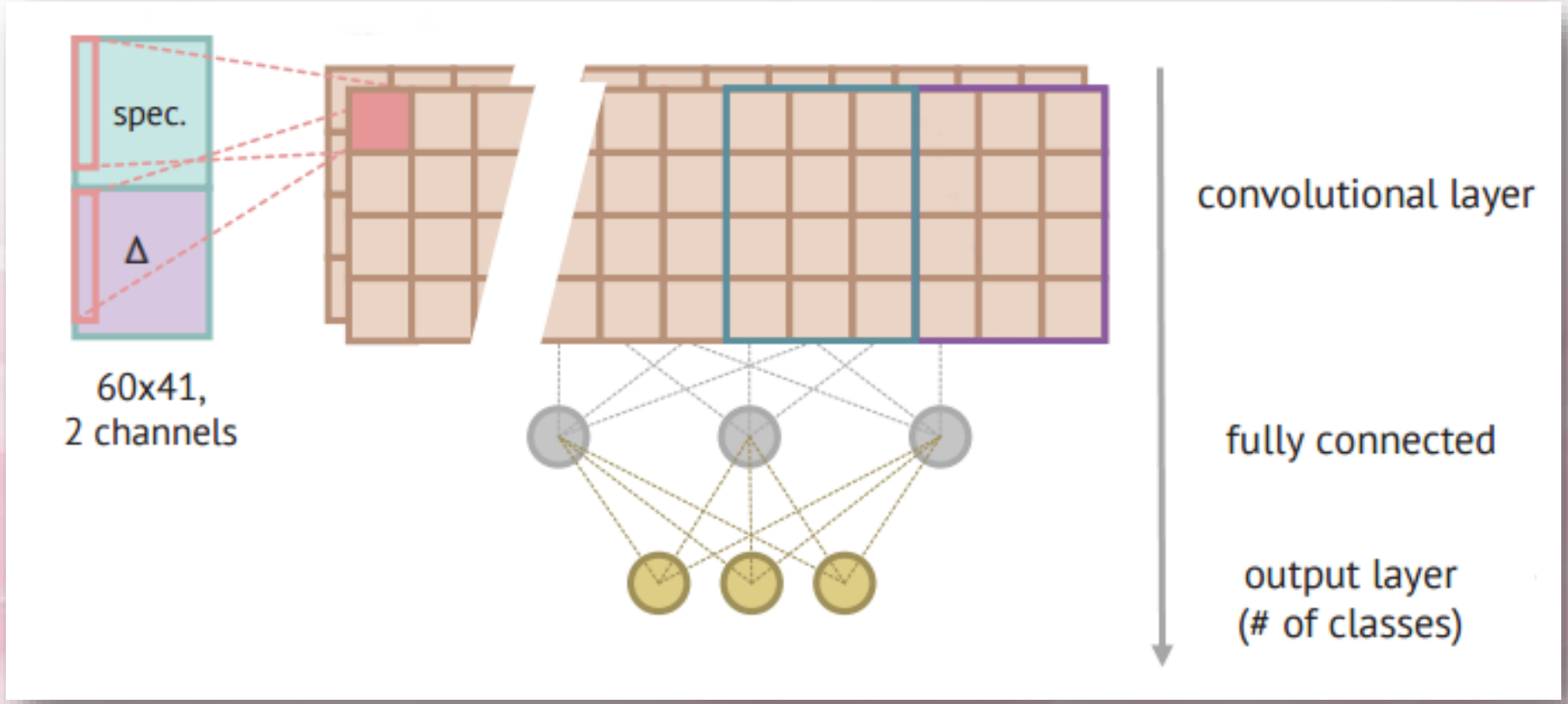
- Thực hiện lặp lại quá trình: tích chập mạng → RELU → Pooling giảm chiều.
- Mục đích: nâng cao kết quả đầu ra.
- Lưu ý: Thực hiện lặp quá trình với 1 số lượng vừa đủ vì lập quá nhiều sẽ làm mất đặc trưng khiến xảy ra underfitting.

Áp dụng vào âm thanh

Lớp Fully Connected:

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.

Áp dụng vào âm thanh



IV. Ứng dụng

Ứng dụng phân loại âm thanh:

- An ninh
  - Rút trích âm thanh trong 1 đoạn video, 1 đoạn hội thoại.
  - Có thể phát triển lên nhận dạng tiếng nói.
- Y học
  - Nhận dạng âm dị thường của các cơ quan trong cơ thể.
  - Phân loại tín hiệu sóng truyền.
- …

V. Tài liệu tham khảo

- Recognizing Sounds (A Deep Learning Case Study) – Arthur Juliani.
- Environmental sound sources classification using neural networks – S.Stoeckle, N.Pah, D.H.Kumar.
- Evironmental sound classification with convolution neural networks – Karol J.Piczak
- Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification - Justin Salamon, Juan Pablo Bello

VI. Kết quả thực nghiệm

- Áp dụng mô hình mạng nơ ron tích chập theo hướng xử lý 5 lớp mạng 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.
- Sử dụng kĩ thuật hold-out, chia dữ liệu làm 3 tập: train (7732 mẫu), validation (1000 mẫu), test (1000 mẫu)

VI. Kết quả thực nghiệm

Phương pháp thực nghiệm

- Áp dụng mạng nơ ron tích chập theo mô hình mà tác giả Karol.J.Piczak đề ra và cải tiến tăng thêm số lượng lớp tích chập theo mô hình của tác giả Justin Salamon và Juan Pablo Bello.
- Cụ thể: mô hình gồm 5 lớp mạng 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.
- Sử dụng kĩ thuật hold-out, chia dữ liệu làm 3 tập: train (7732 mẫu), validation (1000 mẫu), test (1000 mẫu)

VI. Kết quả thực nghiệm

Kết quả thực nghiệm

Độ chính xác tốt nhất đạt được: 78.58%

```
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
```

VI. Kết quả thực nghiệm

So sánh kết quả:

- Tác giả Karol.J.Piczak áp dụng mô hình mình đề ra và đạt được độ chính xác là 73.6%.
- 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%.
- Tác giả Justin Salamon và Juan Pablo Bello thực nghiệm trên mô hình SKM, PiczakCNN và SB-CNN đạt được độ chính xác lần lượt là: 74%, 73% và 73%.
- Tác giả Justin Salamon áp dụng mô hình mình đề ra và đạt được độ chính xác lên đến 84%.

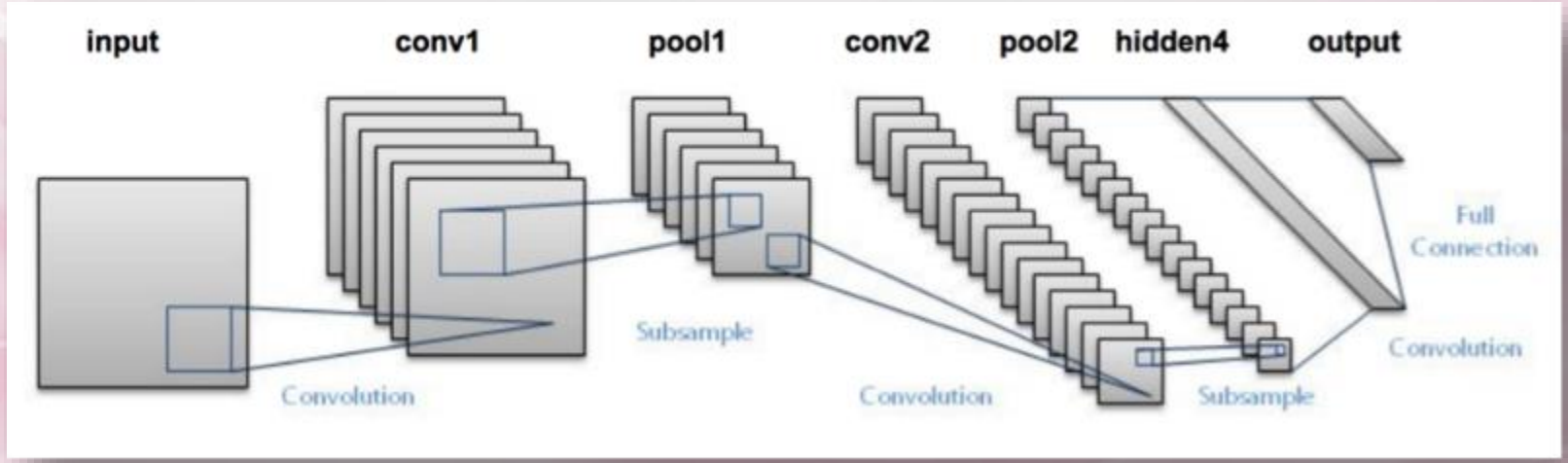
VI. Kết quả thực nghiệm

Nhận xét kết quả:

Với độ chính xác đạt được là 78.58%, ta thấy rằng mô hình mạng nơ ron tích chập và các siêu tham số được sử dụng cho kết quả khả quan.

VI. Demo

- Demo chuẩn hóa tín hiệu âm về dạng phổ đồ.
- Demo training dùng mạng Nơron tích chập.



CẢM ƠN THẦY CÔ VÀ CÁC BẠN ĐÃ CHÚ Ý LẮNG NGHE