

Nhận dạng mẫu (Pattern Recognition)

Mạng học sâu (Deep learning network)

By Hoàng Hữu Việt

Email: viethh@vinhuni.edu.vn

Viện Kỹ thuật và Công nghệ, Đại học Vinh

Vinh, 5/2019

Tài liệu tham khảo

■ Tài liệu chính

[1] Martin T. Hagan, Howard B. Demuth, Mark Hudson Beale.
Neural Network Design, 2nd.

link: hagan.okstate.edu/nnd.html.

[2] Deep Learning with Python, FRANCOIS CHOLLET, Manning,
2018.

■ Tài liệu khác

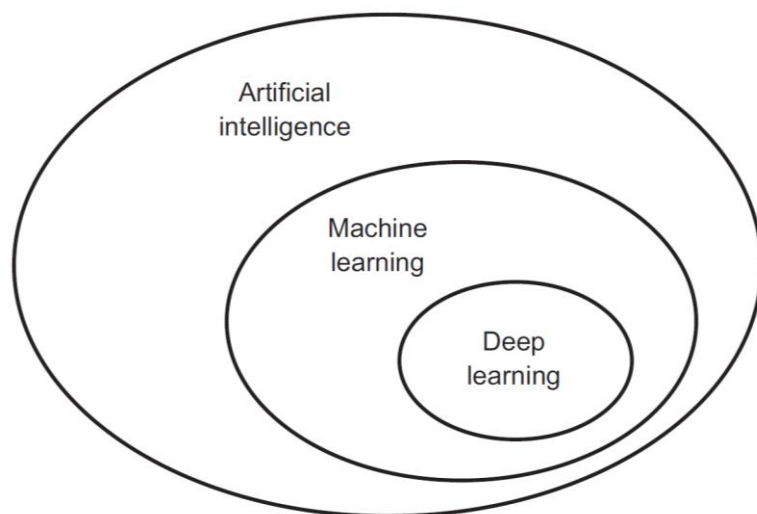
[2] Các nguồn từ internet

Nội dung

- Giới thiệu
- Mạng nhân chập cho nhận dạng ảnh

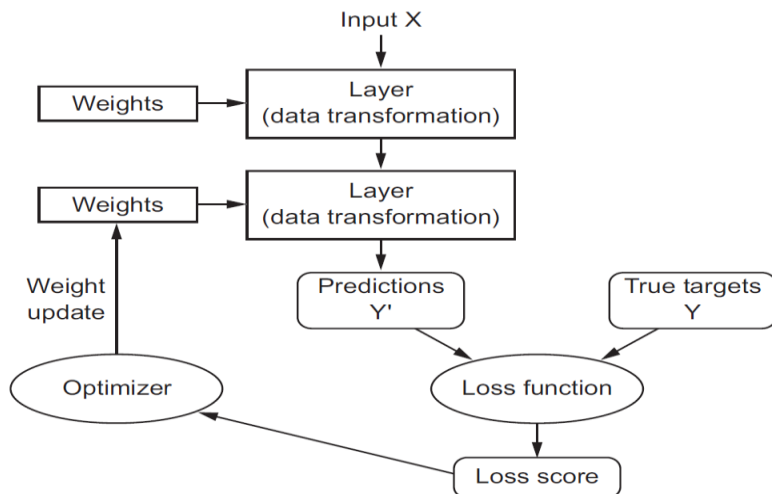
Giới thiệu

- Trí tuệ nhân tạo, học máy và học sâu:



Giới thiệu

- Hàm mất mát (loss function) dùng để điều chỉnh tập tham số của mạng để đạt sự mất mát là bé nhất.



Giới thiệu

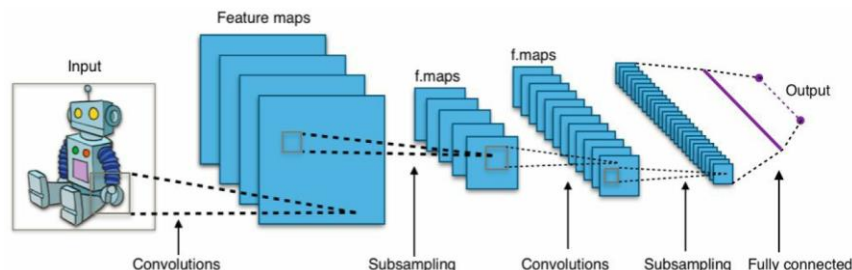
- Những thành tựu dựa trên mạng học sâu trong những năm gần đây:
 - ❑ Near-human-level image classification
 - ❑ Near-human-level speech recognition
 - ❑ Near-human-level handwriting transcription
 - ❑ Improved text-to-speech conversion
 - ❑ Digital assistants such as Google Now and Amazon Alexa
 - ❑ Near-human-level autonomous driving
 - ❑ Improved search results on the web
 - ❑ Ability to answer natural-language questions

Giới thiệu

- Deep learning: 1997?, why now?
 - Hardware
 - Datasets and benchmarks
 - Algorithmic advances

Mạng nhân chập cho nhận dạng ảnh

- Mạng nơ-ron nhân chập (convolutional neural networks - CNN) là một mạng điển hình cho mô hình học sâu được dùng trong các ứng dụng của lĩnh vực thị giác máy tính (computer vision).
 - CNN là một mạng nơ-ron truyền thẳng có nhiều tầng, trong đó tầng nhân chập và tầng pooling được sắp xếp hoán đổi lẫn nhau. Tầng cuối cùng là một tầng kết nối đầy đủ.



Mạng nhân chập cho nhận dạng ảnh

- Ví dụ một mạng CNN nhận dạng các chữ số viết tay:

```
from keras import layers
from keras import
models model = models.Sequential()
model.add(layers.Conv2D(32,(3,3),activation='relu',
                        input_shape=(28,28,1)))
model.add(layers.MaxPooling2D((2,2)))
model.add(layers.Conv2D(64,(3,3),activation='relu'))
model.add(layers.MaxPooling2D((2,2)))
model.add(layers.Conv2D(64,(3,3),activation='relu'))
model.add(layers.Flatten())
model.add(layers.Dense(64,activation='relu'))
model.add(layers.Dense(10,activation='softmax'))
```

Mạng nhân chập cho nhận dạng ảnh

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 26, 26, 32)	320
maxpooling2d_1 (MaxPooling2D)	(None, 13, 13, 32)	0
conv2d_2 (Conv2D)	(None, 11, 11, 64)	18496
maxpooling2d_2 (MaxPooling2D)	(None, 5, 5, 64)	0
conv2d_3 (Conv2D)	(None, 3, 3, 64)	36928
flatten_1 (Flatten)	(None, 576)	0
dense_1 (Dense)	(None, 64)	36928
dense_2 (Dense)	(None, 10)	650

=====

Total params: 93,322
Trainable params: 93,322
Non-trainable params: 0

Mạng nhân chập cho nhận dạng ảnh

■ Huấn luyện, nhận dạng:

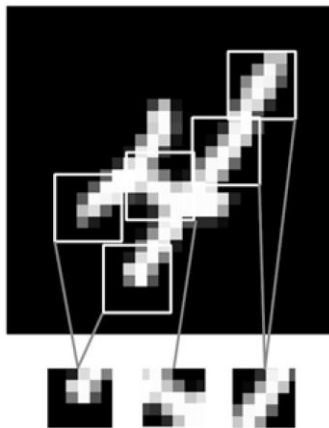
```
from keras.datasets import mnist
from keras.utils import to_categorical
(train_images, train_labels), (test_images, test_labels) =
    mnist.load_data()

train_images = train_images.reshape((60000, 28, 28, 1))
train_images = train_images.astype('float32')/255
test_images = test_images.reshape((10000, 28, 28, 1))
test_images = test_images.astype('float32')/255
train_labels = to_categorical(train_labels)
test_labels = to_categorical(test_labels)
model.compile(optimizer='rmsprop',
              loss='categorical_crossentropy', metrics=['accuracy'])
model.fit(train_images, train_labels, epochs=5, batch_size=64)
test_loss, test_acc = model.evaluate(test_images, test_labels)
```

Mạng nhân chập cho nhận dạng ảnh

■ Khác nhau cơ bản giữa tầng Dense và tầng Conv2D:

- Tầng Dense học các mẫu tổng thể (global patterns).
- Tầng Conv2D học các mẫu cục bộ (local patterns).

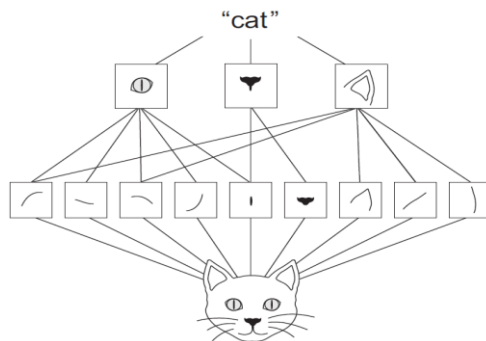


Mạng nhân chập cho nhận dạng ảnh

- Tầng nhân chập có khả năng học các mẫu dịch chuyển:
 - Ví dụ sau khi học một mẫu nào đó ở góc trên bên trái của ảnh, mạng nhân chập có thể nhận ra mẫu đã học có mặt ở bất kỳ vị trí nào khác trong ảnh.
- Tầng kết nối đầy đủ phải học lại các mẫu mới nếu mẫu xuất hiện ở một vị trí mới.
- Mạng nhân chập hiệu quả hơn khi nhận dạng ảnh.

Mạng nhân chập cho nhận dạng ảnh

- Tầng nhân chập có khả năng học phân cấp không gian các mẫu (spatial hierarchies of patterns).
- Tầng 1 của mạng nhân chập sẽ học các mẫu cục bộ nhỏ (ví dụ các cạnh).
- Tầng 2 học các mẫu lớn hơn được tạo ra từ tầng 1.



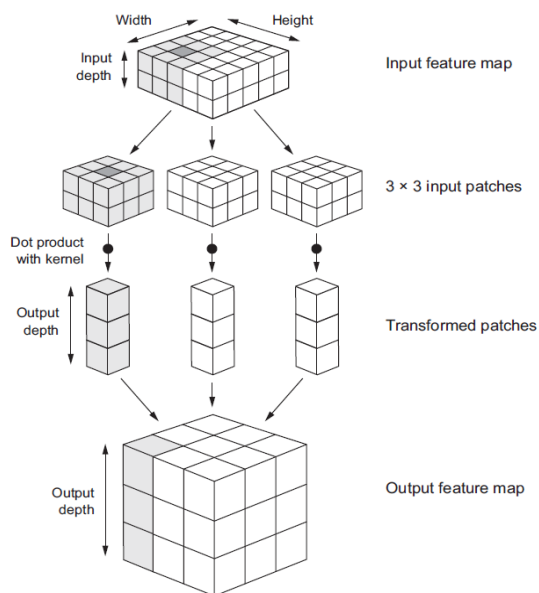
Mạng nhân chập cho nhận dạng ảnh

■ Nguyên lý làm việc của tầng nhân chập:

- Dịch chuyển các cửa sổ kích thước 3×3 hoặc 5×5 trên dữ liệu vào, dừng lại vị trí có thể, và trích ra các khối có kích thước (`window_height`, `window_width`, `input_depth`) của các đặc trưng.
- Mỗi khối sau đó được chuyển thành một vector (`output_depth`) bằng toán tử nhân vô hướng.
- Tất cả các khối sau đó được ghép lại thành một bản đồ đặc trưng đầu ra với kích thước (`height`, `width`, `output_depth`).
- Mỗi vị trí không gian trong bản đồ đặc trưng đầu ra với vị trí tương ứng trong bản đồ đặc trưng đầu vào.

Mạng nhân chập cho nhận dạng ảnh

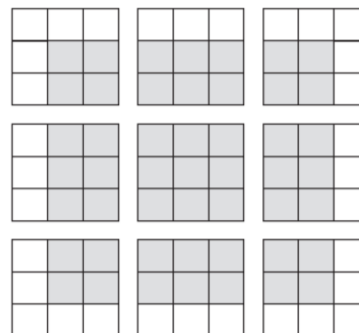
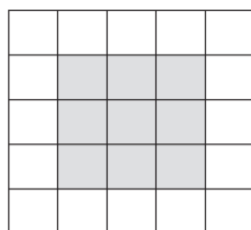
■ Nguyên lý làm việc tầng nhân chập:



Mạng nhân chập cho nhận dạng ảnh

■ Tầng nhân chập - hiệu ứng biên:

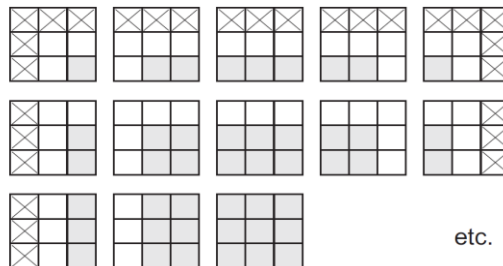
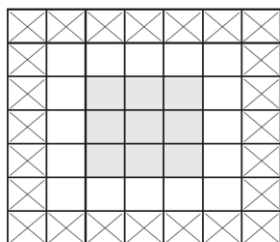
- Xét một bản đồ đặc trưng kích thước 5×5 , tức là có 25 ô (25 tiles).
- Nếu chọn cửa sổ nhân chập kích thước 3×3 , đầu ra của bản đồ đặc trưng có kích thước là 3×3 (tức là $5 \times 5 \rightarrow 3 \times 3$, tương tự $28 \times 28 \rightarrow 26 \times 26$).



Mạng nhân chập cho nhận dạng ảnh

■ Tầng nhân chập - hiệu ứng biên:

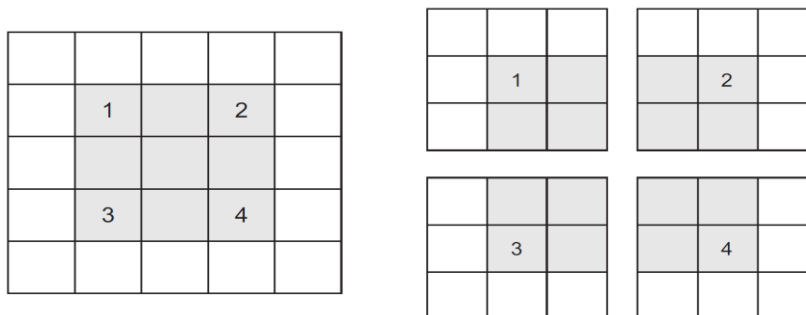
- Nếu muốn kích thước của bản đồ đặc trưng đầu ra giống với kích thước bản đồ đặc trưng đầu vào??
- Bổ sung thêm (padding) một số dòng và một số cột trên mỗi cạnh của bản đồ đặc trưng đầu vào
- Đặt trung tâm của cửa sổ nhân chập nằm ở các ô của các cạnh của bản đồ đặc trưng đầu vào.



Mạng nhân chập cho nhận dạng ảnh

■ Tầng nhân chập - hiệu ứng biên:

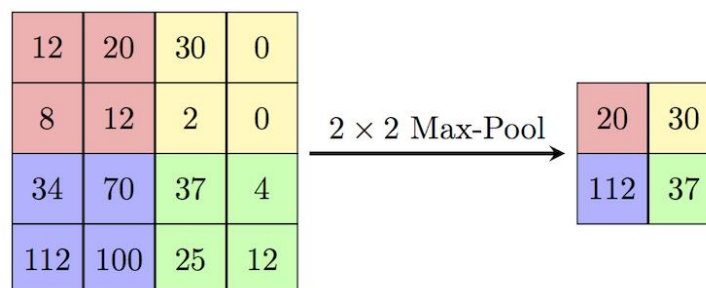
- Một nhân tố khác có thể ảnh hưởng đến kích thước đầu ra là **strides**.
- Khoảng cách giữa hai cửa sổ nhân chập liên tục là một tham số của nhân chập và được gọi là **stride**, ngầm định là 1, có thể thay đổi $\text{stride} = 2$.



Mạng nhân chập cho nhận dạng ảnh

■ Tầng max-pooling:

- Vai trò chính của tầng MaxPooling2D là giảm kích thước bản đồ đặc trưng đầu vào.
- Tầng MaxPooling2D thường dùng kích thước cửa sổ nhân chập là 2×2 với $\text{stride} = 2$ để giảm một nửa kích thước đầu vào.



Bài tập

1. Cho 2 thư mục hws-train và hws-test chứa các ảnh chữ ký của 79 người, trong đó thư mục hws-train chứa ảnh dùng để huấn luyện và thư mục hws-test dùng chứa các ảnh nhận dạng. Tên tệp trong 2 thư mục chứa một dấu '-' và các ký tự trước dấu '-' biểu diễn lớp dữ liệu của ảnh, các ký tự sau dấu '-' là số thứ tự chữ ký của mỗi người. Hãy thiết kế và lập trình một mạng CNN để nhận dạng các chữ ký.
2. Thiết kế một mạng CNN nhận dạng mặt người.