Báo cáo

# Image Classification

## Các bài toán trong Computer Vision

CV (Computer Vision) – là một lĩnh vực làm cho máy tính có thể có khả năng nhìn được như con người, nhìn ở đây có nghĩa là hiểu được hình ảnh, video từ đó có thể làm cơ sở để trả lời các câu hỏi khác như: có gì trong ảnh ? con mèo ở đâu trong bức ảnh? trong ảnh có bao nhiêu con chó, con mèo? … Để trả lời các câu hỏi này thì trong CV sẽ có các bài toán khác nhau. 1 số bài toán cơ bản bao gồm: Image classification, Classification with Localization, Object Detection, Instance Segmentation( Ở báo cáo này sẽ tập trung nói về Image Classification) :

Chart, diagram, box and whisker chart

Description automatically generated

Các bài toán và ứng dụng của CV

* Classification with Localization: Là bài toán phân loại và xác định vị trí của đối tượng
* Object Detection: Là bài toán xác định vị trí của đối tượng
* Instance Segmentation: Là bài toán phân đoạn ảnh, vẽ được đường biên bao quanh đối tượng

|  |  |  |
| --- | --- | --- |
|  | Input | Output |
| Classification | Ảnh chỉ chứa một đối tượng | Gắn label cho đối tượng trong ảnh |
| Localization | Gắn label và boundary cho đối tượng trong ảnh |
| Object Detection | Ảnh có thể chứa nhiều đối tượng | Gắn label và boundary cho các đối tượng trong ảnh |
| Segmentation | Gắn label cho pixel của từng đối tượng trong ảnh |

## Image Classification

Image Classification hay hiểu đơn giản là phân loại hình ảnh là một trong những nhiệm vụ phổ biến trong Computer Vision.

Phân loại các đối tượng là một nhiệm vụ khá dễ dàng đối với chúng ta, nhưng nó là một công việc phức tạp đối với máy móc và là một nhiệm vụ quan trọng trong Computer Vision. Việc kiểm tra và phân loại hình ảnh theo cách thủ công có thể là một công việc lặp lại và dễ dàng với con người, đặc biệt là khi chúng có số lượng lớn (ví dụ 10.000 bức ảnh), khi đó sẽ rất hữu ích nếu chúng ta có thể tự động hóa toàn bộ quy trình này bằng cách sử dụng CV.

Mục tiêu chính của bài toán này đó chính là phân loại một hình ảnh đầu vào (input) thành một nhãn (label) đầu ra (output). Một ví dụ đơn giản từ một bức ảnh đã cho chúng ta cần phân biệt bức ảnh đó là con chó hay con mèo.

Kết quả bài toán phân lớp ảnh có thể áp dụng vào rất nhiều lĩnh vực như phân loại động vật, phân loại biển báo giao thông,… Phân lớp ảnh cũng được cho là là bài toán cơ sở cho một số bài toán khác trong CV. Tuy vậy, trong thực tế bài toán cũng có nhiều thách thức:

* **Đa dạng về góc nhìn**–**Viewpoint variation**: Đối tượng cần phân loại khác nhau khi có góc nhìn khác nhau.
* **Đa dạng về tỉ lệ/ kích thước** – **Scale variation**: Cùng một đối tượng cần phân loại tuy nhiên có thể có kích thước khác nhau.
* **Biến dạng** – **Deformation**: Sự đa dạng hình ảnh của cùng một đối tượng khi bị biến đổi theo các điều kiện khác nhau.
* **Bị che khuất**– **Occlusion**: Đối tượng cần phân loại bị che khuất một phần.
* **Điều kiện chiếu sáng**– **Illumination conditions**: Cùng một ảnh chụp một vật nhưng ở những thiết lập chiếu sáng khác nhau có thể gây sai sót trong quá trình phân loại.
* **Ảnh hưởng bởi bối cảnh** – **Background clutter**: Đối tượng cần phân loại bị nhầm lẫn với môi trường xung quanh.
* **Đa dạng về biến thể trong một nhãn** – **Intra-class variation**: Nhãn cần phân loại có rất nhiều loại ví dụ xe máy thì có xe ga, xe côn, xe số.

## Supervised learning vs Unsupervised learning

Có nhiều thuật toán khác nhau được ứng dụng trong việc phân loại hình ảnh. Các thuật toán này được chia thành hai nhóm chính là Học có giám sát (supervised learning) và Học không giám sát (unsupervised learning).

* Supervised learning là thuật toán dự đoán đầu ra (outcome) của một dữ liệu mới (new input) dựa trên các cặp (input, outcome) đã biết từ trước.  
  Trong Supervised learning chia ra 2 nhóm chính:
  + Classification: các label của input data được chia thành một số hữu hạn nhóm. VD: dự đoán thư spam hay không
  + Regression: các label không được chia thành các nhóm mà là một giá trị thực cụ thể. VD: dự đoán tuổi
* Unsupervised learning là thuật toán mà không biết được outcome hay nhãn mà chỉ có dữ liệu đầu vào.  
  Trong Unsupervised learning chia ra 2 nhóm chính:
  + Clustering: là bài toán phân nhóm toàn bộ dữ liệu thành các nhóm nhỏ dựa trên sự liên quan giữa các dữ liệu trong mỗi nhóm.
  + Non-clustering : là bài toán khi chúng ta muốn khám phá ra một quy luật dựa trên nhiều dữ liệu cho trước.

Thuật toán sử dụng trong báo cáo là K-NN và Neural Network (Supervised learning) trên cơ sở tập dữ liệu chữ số viết tay MNIST.

# Neural Network

Neural Network (còn gọi là ANN hay SNN) mô phỏng lại hành vi của não người, cho phép các chương trình máy tính nhận ra và giải quyết các vấn đề phổ biến trong lĩnh vực AI, Machine Learning và Deep learning. Tên và cấu trúc của chúng được lấy cảm hứng từ não người, bắt chước cách các tế bào thần kinh sinh học truyền tín hiệu cho nhau.

Mạng nơ-ron dựa trên dữ liệu đào tạo để tìm hiểu và cải thiện độ chính xác của chúng theo thời gian, là một thuật toán supervised learning. Một khi các thuật toán học tập này được tinh chỉnh về độ chính xác, chúng sẽ là công cụ mạnh mẽ trong khoa học máy tính và trí tuệ nhân tạo, cho phép chúng ta phân loại và phân cụm dữ liệu với tốc độ cao.

Một Neural Network nổi tiếng là thuật toán tìm kiếm của Google.

## Cách hoạt động

Mạng nơ-ron cấu tạo từ 3 kiểu tầng chính là Input Layer, Hidden Layer, Output Layer; trong đó Input Layer và Output Layer chỉ có một tầng nhưng Hidden Layer ở giữa chúng có thể bao gồm nhiều tầng ẩn khác nhau.

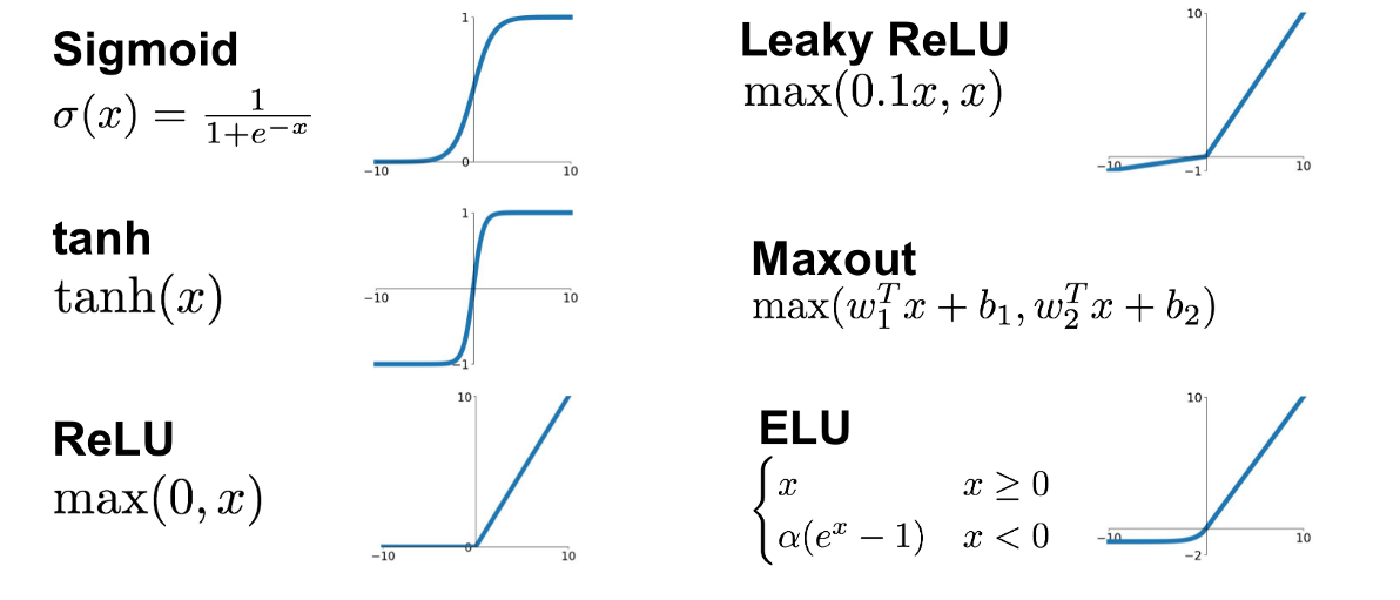
Diagram

Description automatically generated

Mỗi tầng được cấu tạo từ các nút (node). Tại mỗi tầng, số nút có thể khác nhau tuỳ điều kiện bài toán.

Mỗi nút riêng lẻ như một mô hình hồi quy tuyến tính (Linear Regression) của riêng nó, bao gồm dữ liệu đầu vào (input data), trọng số (weights), độ chệch (bias) và đầu ra (output). Công thức sẽ giống như sau: ∑wixi + bias = w1x1 + w2x2 + w3x3 + bias

Sau đó, tại mỗi nút, áp dụng activation function để cho ra kết quả, và cũng coi như đầu vào cho các nút ở các layer tiếp theo.



Một số hàm Activation Function

# Keras

Keras là một open source cho Neural Network được viết bởi ngôn ngữ Python. Nó là một library được phát triển vào năm 2005 bởi Francois Chollet, là một kỹ sư nghiên cứu Deep Learning. Keras có thể sử dụng chung với các thư viện nổi tiếng như Tensorflow, CNTK, Theano. Một số ưu điểm của Keras như:

* Dễ sử dụng, dùng đơn giản hơn Tensor, xây dựng model nhanh.
* Run được trên cả CPU và GPU.
* Hỗ trợ xây dựng CNN , RNN

Xây dựng model bằng Sequential function trong Keras:

* Khởi tạo models bằng Sequential()
* Thêm các chức năng với add()
* Convolution2D : Tạo Convolutional Layer (trong CNN)
* Pooling Layers: (MaxPooling2D,AvergaPooling1D,2D,…): giảm param khi train
* Dense(): Tạo các layer neural network
* Compile(): Training model với các option về optimizer (chọn thuật toán training), loss\_function, metrics (đánh giá độ chính xác của thuật toán),…
* Fit(): Đưa data train, test vào để training với các tham số: batch-size (số dữ liệu dùng mỗi lần cập nhật trọng số), iterations (số vòng lặp), epochs (một lần duyệt qua hết tập train)

# Convolutional Neural Network

## Phép tính Convolution (Tích chập)

Kí hiệu phép tính Convolution là ⊗ , Y = X ⊗ Z

Với Z là ma trận kernel k\*k (k thường là số lẻ)

Với mỗi phần tử xij ​ trong ma trận X lấy ra một ma trận có kích thước bằng kích thước của kernel W có phần tử xij ​ làm trung tâm (đây là vì sao kích thước của kernel thường lẻ) gọi là ma trận A. Sau đó tính tổng các phần tử của phép tính element-wise của ma trận A và ma trận W, rồi viết vào ma trận kết quả Y.

Ví dụ khi tính tại x22​ (ô khoanh đỏ trong hình), ma trận A cùng kích thước với W, có x22 ​ làm trung tâm có màu nền da cam như trong hình. Sau đó tính y11 = x11\*w11 + x12\*w12 + x13\*w13 + x21\*w21 + x22\*w22 + x23\*w23 + x31\*w31 + x32\*w32 + x33\*w33 ​= 4. Và làm tương tự với các phần tử còn lại trong ma trận.

A picture containing text, white

Description automatically generated

Phép tích chập

Bình thường khi tính thì sẽ bỏ qua các phần tử ở viền ngoài, vì không tìm được ma trận A ở trong X với tâm là các phần tử ở viền ngoài, ví dụ tại x11 (do đó có thể sử dụng Padding)

### Padding

Sau mỗi lần thực hiện phép tính convolution xong thì kích thước ma trận Y đều nhỏ hơn X. Tuy nhiên giờ ta muốn ma trận Y thu được có kích thước bằng ma trận X, phải tìm cách xử lý các viền => Thêm các giá trị 0 ở viền của ma trận X => Khi convolution với kernel, ma trận Y cho ra kích thước giống với X đầu vào.

Phép tính này gọi là convolution với padding=1. Padding=k nghĩa là thêm k vector 0 vào mỗi phía của ma trận.

### Stride

Bắt đầu từ vị trí đầu tiên​ sau đó nhảy Stride bước theo chiều dọc và ngang cho đến hết ma trận X. Kích thước của ma trận Y đã giảm đi đáng kể so với ma trận X.

### Công thức tổng quát

Công thức tổng quát cho phép tính convolution của ma trận X kích thước m\*n với kernel kích thước k\*k, stride = s, padding = p ra ma trận Y kích thước:

## Convolutional Neural Network

Tích chập được ứng dụng phổ biến trong lĩnh vực thị giác máy tính và tạo ra mạng nơ ron tích chập. Thông qua các phép tích chập, các đặc trưng chính từ ảnh được trích xuất và truyền vào các tầng tích chập (layer convolution). Mỗi một tầng tích chập sẽ bao gồm nhiều đơn vị mà kết quả ở mỗi đơn vị là một phép biến đổi tích chập từ layer trước đó thông qua phép nhân tích chập với bộ lọc.

Về cơ bản thiết kế của một mạng nơ ron tích chập 2 chiều có dạng như sau:

INPUT -> [[CONV -> RELU]\*N -> POOL?]\*M -> [FC -> RELU]\*K -> FC

Trong đó:

* INPUT: Tầng đầu vào
* CONV: Tầng tích chập
* RELU: Tầng kích hoạt. Thông qua hàm kích hoạt (activation function), thường là ReLU hoặc LeakyReLU để kích hoạt phi tuyến
* POOL: Tầng tổng hợp, thông thường là Max pooling hoặc có thể là Average pooling dùng để giảm chiều của ma trận đầu vào.
* FC: Tầng kết nối hoàn toàn. Thông thường tầng này nằm ở sau cùng và kết nối với các đơn vị đại diện cho nhóm phân loại.

Các kí hiệu []N, []M hoặc []\*K ám chỉ các khối bên trong [] có thể lặp lại nhiều lần liên tiếp nhau với N, M, K là số lần lặp lại. Kí hiệu -> đại diện cho các tầng liền kề nhau mà tầng đứng trước sẽ làm đầu vào cho tầng đứng sau. Dấu ? sau POOL để thể hiện tầng POOL có thể có hoặc không sau các khối tích chập.

Như vậy ta có thể thấy một mạng nơ ron tích chập về cơ bản có 3 quá trình khác nhau:

* Quá trình tích chập (convolution): Thông qua các tích chập giữa ma trận đầu vào với bộ lọc để tạo thành các đơn vị trong một tầng mới. Quá trình này có thể diễn ra liên tục ở phần đầu của mạng và thường sử dụng kèm với hàm kích hoạt ReLU. Mục tiêu của tầng này là trích xuất đặc trưng hai chiều.
* Quá trình tổng hợp (max pooling): Các tầng càng về sau khi trích xuất đặc trưng sẽ cần số lượng tham số lớn do chiều sâu được quy định bởi số lượng các kênh ở các tầng sau thường tăng tiến theo cấp số nhân. Điều đó làm tăng số lượng tham số và khối lượng tính toán trong mạng nơ ron. Do đó để giảm tải tính toán chúng ta sẽ cần giảm kích thước các chiều của khối ma trận đầu vào hoặc giảm số đơn vị của tầng. Vì mỗi một đơn vị sẽ là kết quả đại diện của việc áp dụng 1 bộ lọc để tìm ra một đặc trưng cụ thể nên việc giảm số đơn vị sẽ không khả thi. Giảm kích thước khối ma trận đầu vào thông qua việc tìm ra 1 giá trị đại diện cho mỗi một vùng không gian mà bộ lọc đi qua sẽ không làm thay đổi các đường nét chính của bức ảnh nhưng lại giảm được kích thước của ảnh. Do đó quá trình giảm chiều ma trận được áp dụng. Quá trình này gọi là tổng hợp nhằm mục đích giảm kích thước dài, rộng.   
  Trong một số model người ta dùng convolutional layer với stride > 1 để giảm kích thước dữ liệu thay cho pooling layer  
  Có 2 loại pooling layer phổ biến là Max pooling và Average pooling
* Quá trình kết nối hoàn toàn (fully connected): Sau khi đã giảm kích thước đến một mức độ hợp lý, ma trận cần được trải phẳng (flatten) thành một vector và sử dụng các kết nối hoàn toàn giữa các tầng. Quá trình này sẽ diễn ra cuối mạng CNN và sử dụng hàm kích hoạt là ReLU. Tầng kết nối hoàn toàn cuối cùng (fully connected layer) sẽ có số lượng đơn vị bằng với số classes và áp dụng hàm kích hoạt là softmax nhằm mục đích tính phân phối xác xuất.

Diagram, engineering drawing

Description automatically generated

# Image segmentation

## Bài toán

Image Segmentation cũng có chung mục tiêu như object detection là phát hiện ra vùng ảnh chứa vật thể và gán nhãn phù hợp cho chúng. Tuy nhiên tiêu chuẩn về độ chính xác của Image Segmentation ở mức cao hơn so với Object Detection khi nó yêu cầu nhãn dự báo đúng tới từng pixel.

Image Segmentation yêu cầu về mức độ chi tiết cao hơn do đó thuật toán giúp ta hiểu được nội dung của một bức ảnh ở mức độ sâu hơn khi chúng ta biết được đồng thời: Vị trí của vật thể trong ảnh, hình dạng của vật thể và từng pixel nào thuộc về vật thể nào.

Một số ứng dụng của Image Segmentation:

* Y học: hình dạng của các tế bào ung thư là một trong những yếu tố quyết định độ ác tính của bệnh, nên ta cần image segmentation để biết được chính xác hình dạng của các tế bào ung thư để có các chẩn đoán xác định. Object detection ở đây không đủ giải quyết được vấn đề.
* Xe tự lái: Một hệ thống xe tự hành cần phải xác định chính xác các vật thể xuất hiện khi tham gia giao thông như người, đèn tín hiệu, biển báo, vạch kẻ đường, xe cộ
* Xử lý ảnh vệ tinh: Các vệ tinh quay quanh trái đất sẽ liên tục thu thập hình ảnh bề mặt trái đất ở những vùng khác nhau. Từ các bức ảnh chụp vệ tinh, mô hình Image Segmentation sẽ phân đoạn hình ảnh thành tuyến đường, khu phố, biển cả, cây cối,….

Image Segmentation nếu được huấn luyện theo bài toán học có giám sát trong thị giác máy tính thì sẽ yêu cầu gán nhãn cho ảnh. Input của bài toán là một bức ảnh và output là một ma trận mask mà giá trị của từng pixel đã được gãn nhãn trên đó.

Có 2 bài toán Image Segmentation chính:

* Semantic segmentation: Thực hiện segment với từng lớp khác nhau, ví dụ: tất cả người là 1 lớp, tất cả ô tô là 1 lớp.
* Instance segmentation: Thực hiện segment với từng đối tượng trong một lớp. Ví dụ có 3 người trong ảnh thì sẽ có 3 vùng segment khác nhau cho mỗi người.