**Phần 1\_1\_Phương pháp biểu diễn dữ liệu và nhãn**

**Notes:**

* *Về format cấu trúc soạn như sau:*
  + *Lý thuyết…*
  + *Bộ code mấu/ ví dụ …*
  + *Ứng dụng (nếu có)...*
* *Mems làm nhớ note tên để mn dễ contact*

[**I. Nội dung chính 1**](#_tqtybmuo0zl3)

[**II. Nội dung biên soạn chi tiết 1**](#_260io44fztum)

### 

### **I. Nội dung chính**



Về dạng bài toán hay công việc (tasks): **phân loại**, **hồi quy**, **phát hiện**, **phân** **đoạn**, **sinh ảnh**. Cho mỗi dạng bài toán, cần hiểu rõ:

* + (1) Phương pháp biểu diễn dữ liệu và nhãn
    - Ví dụ, bài toán **phân loại**:
      * Dữ liệu truyền qua mô hình:
        + khi dùng conv2D, có hình dạng: (N, H, W, C), dùng channel-last hay (N, C, H, W), dùng channel-first;
        + khi dùng conv3D, có hình dạng: (N, D, H, W, C), etc
      * cần biết mã hoá nhãn: dạng văn bản, dạng chỉ số, dạng one-hot là gì?
    - Ví dụ, bài toán **phân đoạn**:
      * mặt nạ (mask) là gì?
    - Ví dụ, bài toán **phát hiện**:
      * boundingbox là gì và bằng biểu diễn như thế nào cho một batch

—------------------------------------------------------------------------------------------------------------------------

### **II. Nội dung biên soạn chi tiết**

**1. Phân Loại:**

***- Phương pháp biểu diễn dữ liệu:*** Dữ liệu thường là hình ảnh, có thể được biểu diễn dưới dạng tensor với các kích thước nhất định. Ví dụ, khi sử dụng Convolutional Neural Networks (CNN), dữ liệu có thể có hình dạng (N, H, W, C) với N là số lượng ảnh trong một batch, H và W là chiều cao và chiều rộng của ảnh, và C là số lượng kênh màu (channel). Có thể sử dụng cả định dạng channel-last (NHWC) hoặc channel-first (NCHW).

The choice of format may depend on the specific deep learning framework being used. For example, TensorFlow typically uses the channel-last format, while PyTorch uses the channel-first format. It's important to check the documentation of the specific framework or library to determine the expected input format for conv2d and conv3d functions.

***- Phương pháp biểu diễn nhãn:*** Nhãn thường được biểu diễn dưới dạng one-hot encoding. Ví dụ, nếu có K lớp, mỗi nhãn sẽ là một vector có K chiều, trong đó chỉ có một chiều có giá trị là 1, tương ứng với lớp của ảnh, và các chiều còn lại là 0.

**2. Hồi Quy:**

- Phương pháp biểu diễn dữ liệu: Dữ liệu đầu vào thường là các đặc trưng số hoặc hình ảnh, được biểu diễn dưới dạng tensor tùy thuộc vào loại mô hình.

- Phương pháp biểu diễn nhãn: Nhãn thường là giá trị số, có thể là một giá trị duy nhất hoặc một loạt các giá trị.

**3. Phát Hiện (Object Detection):**

- Phương pháp biểu diễn dữ liệu: Dữ liệu thường bao gồm hình ảnh và thông tin bounding box. Dữ liệu có thể được biểu diễn dưới dạng tensor (N, H, W, C) hoặc (N, C, H, W), tùy thuộc vào sự chọn lựa giữa channel-last và channel-first.

- Phương pháp biểu diễn nhãn: Bounding box thường được biểu diễn bằng cách chỉ định tọa độ của nó trên ảnh (ví dụ, (x\_min, y\_min, x\_max, y\_max)). Nhãn có thể được mã hóa thông qua việc kết hợp thông tin về lớp của đối tượng và tọa độ bounding box.

**4. Phân Đoạn (Semantic Segmentation):**

- Phương pháp biểu diễn dữ liệu: Dữ liệu thường là hình ảnh và mặt nạ (mask), trong đó mỗi pixel trong mặt nạ được gán một nhãn tương ứng với loại đối tượng.

- Phương pháp biểu diễn nhãn: Mặt nạ (mask) được sử dụng để chỉ ra vị trí và đối tượng của từng pixel trong hình ảnh.

**5. Sinh Ảnh (Image Generation)**:

- Phương pháp biểu diễn dữ liệu: Dữ liệu thường được biểu diễn dưới dạng vector đặc trưng hoặc tensor, tùy thuộc vào mô hình đang sử dụng.

- Phương pháp biểu diễn nhãn: Trong trường hợp này, không có nhãn cụ thể để biểu diễn. Mô hình được huấn luyện để sinh ra dữ liệu mới từ dữ liệu đầu vào đã cho.

Bouding box for batch

There are three possible representations for batch of bounding boxes.

1. An array of shape (B, R, 4).
2. A coordinate array and a batch index array. They are shape (R', 4) and (R',) respectively. This representation is used for chainer.functions.roi\_pooling\_2d.
3. List of (R, 4) arrays

The convention on the conditions for selecting representations had not been discussed extensively yet, and I would like to discuss it here.

First of all, I would like to summarize examples found in the code.

* For a function that takes list of images as input and returns bounding box (List[img] -> BBox), list of (R, 4) arrays is used. This is found in predict of detection links.
* For a function that takes batch of image arrays as input and returns bounding box (BCHW -> BBox), batch of bounding box arrays (i.e. an array of shape (B, R, 4)) is used. This is found in SSD.\_\_call\_\_.

Here are rules that I am thinking of.

* When number of bounding boxes per image is fixed, use (B, R, 4).
* When number of bounding boxes per image is varying, use (R', 4) and (R',).
* When it is output of a function with list of images as input, return list of (R, 4).

(R', 4) is the coordinates of bounding boxes and (R') is the indices of corresponding image in input batch