

## Báo cáo tuần 3

**Thực tập sinh:** Trần Văn Trung.

### 1. L1 vs L2 và các trường hợp sử dụng.

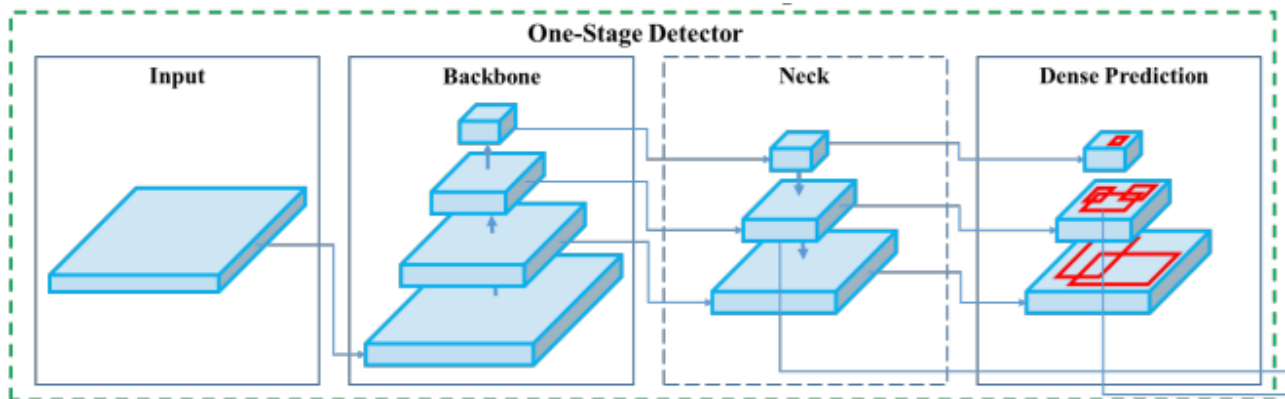
- Việc lựa chọn Lambda rất quan trọng trong sử dụng L1, L2 để tránh over-fitting, nếu Lambda quá lớn lại rất dễ dẫn đến under-fitting.

	L1 (Lasso Regression)	L2 (Ridge Regression)
Cách hiểu	- Cộng thêm một tham số vào hàm mất mát, tham số đó là tổng giá trị tuyệt đối các trọng số $\mathbf{w}$ , $\lambda$ là một siêu tham số mà chúng ta phải chọn giá trị, giá trị $\lambda$ lớn sẽ ưu tiên những trọng số nhỏ hơn trọng số lớn.	- Tham số tương ứng được sử dụng trong L2 là tổng bình phương các trọng số $\mathbf{w}$ . Lambda hoạt động tương tự trong L1.
Công thức hàm mất mát	<div style="text-align: center;"> <p>L1 Regularization</p> <math display="block">\text{Cost} = \sum_{i=0}^N (y_i - \sum_{j=0}^M x_{ij} W_j)^2 + \lambda \sum_{j=0}^M  W_j </math> <p>L2 Regularization</p> <math display="block">\text{Cost} = \underbrace{\sum_{i=0}^N (y_i - \sum_{j=0}^M x_{ij} W_j)^2}_{\text{Loss function}} + \lambda \underbrace{\sum_{j=0}^M W_j^2}_{\text{Regularization Term}}</math> </div>	
Trường hợp sử dụng	<ul style="list-style-type: none"> <li>- Sử dụng trong các trường hợp có nhiều thuộc tính, nhiều features, và có nhiều features có độ tin cậy và độ ảnh hưởng không cao, L1 đưa các trọng số của features đó giảm dần về 0 hoặc loại bỏ luôn features đó.</li> <li>- Tính chất lựa chọn đặc trưng rất cao (features selection).</li> <li>- L1 thường được sử dụng trong các mô hình bài toán có nhiều features và không hiệu quả trong các bài toán có <math>\mathbf{w}</math> là non-sparse.</li> <li>- Trong các bài toán có ít features, L1 cũng hoạt động tốt.</li> <li>- Dùng trong các bài toán muốn tính toán được sự ảnh hưởng của các thuộc tính khác nhau bằng việc thay đổi Lambda để loại bỏ các thuộc tính khác nhau.</li> </ul>	<ul style="list-style-type: none"> <li>- L2 hiệu quả hơn L1 trong việc giải quyết các bài toán mà có <math>\mathbf{w}</math> là non-sparse (có nhiều phần tử khác 0).</li> <li>- Tính chất lựa chọn thuộc tính (features selection) của L2 bị giảm đi.</li> <li>- L2 được sử dụng trong các bài toán NN để tránh over-fitting và được gọi là weight decay (tiêu biến).</li> <li>- Dùng trong các bài toán muốn tìm kiếm kết quả tối ưu.</li> </ul>

## 2. Yolo V3 vs Yolo v4. SSD vs Yolo.

- Yolo v3 vs Yolo V4:

Về kiến trúc:



Kiến trúc Yolo v4

YOLO v1:

- Sử dụng DarkNet framework.
- Thêm 4 lớp Convolutional và 2 lớp FC vào mô hình DarkNet cải tiến.
- Có 24 lớp Conv và 2 lớp FC.

YOLO v2:

- Batch Normalization: giảm sự thay đổi giá trị unit trong hidden layer, do đó sẽ cải thiện được tính ổn định của neural network.
- Higher Resolution Classifier: Kích thước đầu vào trong YOLO v2 được tăng từ  $224 \times 224$  lên  $448 \times 448$ .
- Anchor boxes: dự đoán bounding box và được thiết kế cho tập dữ liệu đã cho sử dụng clustering.
- Fine-Grained Features: YOLO v2 chia ảnh thành  $13 \times 13$  grid cells, do đó có thể phát hiện được những object nhỏ hơn, đồng thời cũng hiệu quả với các object lớn.
- Multi-Scale Training: YOLO v1 có điểm yếu là phát hiện các đối tượng với các kích cỡ đầu vào khác nhau. Điều này được giải quyết bằng YOLO v2, nó được train với kích thước ảnh ngẫu nhiên trong khoảng  $320 \times 320$  đến  $608 \times 608$ .
- Darknet 19: YOLO v2 sử dụng Darknet 19 với 19 convolutional layers, 5 max pooling layers và 1 softmax layer.

YOLO v3: phát hiện, phân loại chính xác đối tượng, và được xử lý thời gian thực.

- Bounding Box Predictions: cung cấp score mỗi bounding boxes sử dụng logistic regression.
- Class Predictions: sử dụng logistic classifiers cho mọi class thay vì softmax.
- Feature Pyramid Networks(FPN).
- Darknet-53

YOLO v4:

- Mô hình phát hiện đối tượng hiệu quả và mạnh mẽ: Tạo CNN hoạt động theo thời gian thực trên GPU thông thường.
- Dễ dàng đào tạo và triển khai hệ thống phát hiện đối tượng

- Đã xác minh tầm ảnh hưởng của các phương pháp phát hiện đối tượng **Bag-of freebies and Bag-of-Specials methods** trong quá trình đào tạo.
- Sửa đổi các phương pháp tiên tiến nhất và làm cho chúng hiệu quả hơn và phù hợp hơn cho việc đào tạo GPU đơn, bao gồm CBN, PAN, SAM, v.v.

Về hiệu quả:

- Yolo V4 cải thiện gấp đôi hiệu năng của EfficientNet.
- Cải thiện AP và FPS của YOLOv3 lần lượt là 10% và 12%.

