TÓM TẮT CHUYÊN ĐỀ DEEP COMPUTER VISION

1 KIẾN TRÚC VÀ CÁC LỚP CƠ BẢN CỦA MẠNG CNN

Mạng nơ-ron tích chập (CNN) là xương sống của Deep Learning trong xử lý hình ảnh, giúp tự động trích xuất các đặc trưng phân cấp từ dữ liệu ảnh thô. [Image of Convolutional Neural Network architecture]

1.1 Lớp Tích chập (Convolutional Layer)

Lớp này thực hiện phép tích chập giữa ảnh đầu vào (*Input*) và một tập hợp các bộ lọc (*Filters/Kernels*), tạo ra **bản đồ đặc trưng (Feature Maps)**.

- **Bộ lọc (K):** Các ma trận nhỏ chứa các trọng số học được. Mỗi bộ lọc đóng vai trò như một bộ phát hiện đặc trưng cụ thể. Các bộ lọc ở lớp nông (early layers) học các đặc trưng cơ bản (cạnh, góc), trong khi các lớp sâu hơn học các đặc trưng phức tạp (mắt, bánh xe, khuôn mặt).
- **Phép Tích chập:** Là phép nhân chập giữa bộ lọc và các vùng nhỏ của ảnh. Đầu ra của một feature map M tại vị trí (i, j) được tính bằng:

$$M_{i,j} = \sum_{x=0}^{K_w-1} \sum_{y=0}^{K_h-1} I_{i \cdot S+x, j \cdot S+y} \cdot K_{x,y} + b$$

  Trong đó: I là ảnh đầu vào, K là kernel (kích thước $K_w \times K_h$), S là stride, và b là bias (độ chệch). Bias b cho phép dịch chuyển đường kích hoạt.

- **Kích thước Đầu ra (O):** Kích thước không gian (chiều rộng hoặc chiều cao) của bản đồ đặc trưng đầu ra sau khi tích chập được tính theo công thức:

$$O = \left\lfloor \frac{I - K + 2P}{S} \right\rfloor + 1$$

  Với I: Kích thước đầu vào, K: Kích thước kernel, P: Padding, S: Stride.

- **Padding (Đệm):**
  1. **Valid:** Không đệm ($P = 0$). Kích thước đầu ra nhỏ hơn đầu vào.
  2. **Same:** Đệm sao cho kích thước đầu ra xấp xỉ kích thước đầu vào. Thường dùng cho các mạng muốn duy trì thông tin không gian.

1.2 Lớp Pooling (Pooling Layer)

Mục tiêu là giảm kích thước không gian (*downsampling*), giúp giảm số lượng tham số, giảm chi phí tính toán, và tạo ra sự bất biến với dịch chuyển nhỏ.

- **Max Pooling:** Chọn giá trị lớn nhất trong cửa sổ. Max Pooling thường dùng 2×2 với stride 2. Đây là một phương pháp giảm mẫu (subsampling) hiệu quả mà không làm mất quá nhiều thông tin quan trọng.

- **Average Pooling:** Lấy giá trị trung bình trong cửa sổ. Thường được sử dụng như một giải pháp thay thế cho Max Pooling hoặc trong các lớp cuối cùng.
- **Global Average Pooling (GAP):** Lấy giá trị trung bình trên toàn bộ feature map (kích thước $W \times H \times C$) để tạo ra một vector $1 \times 1 \times C$. Thường thay thế các lớp FC cuối cùng trong các kiến trúc hiện đại (ví dụ: GoogLeNet, ResNet).

1.3 Lớp Kích hoạt (Activation Layer)

Áp dụng hàm phi tuyến tính sau mỗi phép tích chập (hoặc chuẩn hóa hàng loạt) để cho phép mạng học các ánh xạ phi tuyến tính và phức tạp.

- **ReLU (Rectified Linear Unit):** Là hàm kích hoạt tiêu chuẩn cho các lớp ẩn, giúp giải quyết vấn đề gradient biến mất (*vanishing gradient*) hiệu quả hơn các hàm truyền thống (như Sigmoid hay Tanh). Công thức:
$$f(z) = \max(0, z)$$
- **Các biến thể:** Leaky ReLU, Parametric ReLU (PReLU), và ELU được sử dụng để giải quyết vấn đề "chết" của nơ-ron (dying ReLU) khi đầu vào là số âm.

1.4 Lớp Kết nối Đầy đủ (Fully Connected Layer - FC)

Các lớp FC nằm ở cuối mạng, nhận vector đặc trưng 1D (đã được làm phẳng - *flattened*) từ các lớp tích chập trước đó và sử dụng chúng để thực hiện phân loại cuối cùng (thường dùng hàm softmax cho đầu ra đa lớp).

2 CÁC KIẾN TRÚC CNN NỔI BẬT VÀ PRETRAINED CNN

2.1 Lịch sử Phát triển các Kiến trúc

- **AlexNet (2012):** Đặt nền móng cho CNN sâu, sử dụng GPU, ReLU và **Dropout**.
- **VGGNet (2014):** Nổi tiếng với sự đơn giản và đồng nhất, chỉ sử dụng các lớp tích chập 3×3 và 1×1. Độ sâu mạng rất lớn (lên đến 19 lớp), là kiến trúc tham khảo cho việc trích xuất đặc trưng.
- **GoogLeNet/Inception (2014):** Giới thiệu **Inception Module** (Module Khởi tạo). Module này thực hiện các phép tích chập 1×1, 3×3, 5×5 và max pooling song song trên cùng một đầu vào, sau đó nối (concatenate) các kết quả lại. Phép tích chập 1×1 được dùng để giảm chiều sâu (dimensionality reduction), giúp giảm chi phí tính toán.
- **ResNet (2015):** Giải quyết vấn đề mạng quá sâu bị suy giảm hiệu suất (*degradation*). ResNet sử dụng **kết nối bỏ qua (skip connections)** hoặc **khối dư thừa (residual blocks)**. Khối dư thừa học phần còn lại ($F(x)$) thay vì hàm đầy đủ ($H(x)$), trong đó:
$$H(x) = F(x) + x$$
Kết nối x trực tiếp giúp gradient dễ dàng truyền ngược trở lại qua nhiều lớp, cho phép xây dựng mạng với hơn 100 lớp.
- **FCN (Fully Convolutional Networks):** Kiến trúc chỉ sử dụng các lớp tích chập, loại bỏ các lớp FC, cho phép chấp nhận đầu vào kích thước bất kỳ và là nền tảng của Segmentation.

2.2 Mạng CNN Huấn luyện Sẵn (Pretrained CNNs)

Khái niệm cốt lõi là **Học Chuyển giao (Transfer Learning)**.

- **Giá trị:** Các mô hình đã được huấn luyện trên các bộ dữ liệu quy mô lớn (như ImageNet) đã học được các đặc trưng thị giác chung.
- **Các chiến lược chính:**
  1. **Trích xuất Đặc trưng (Feature Extraction):** Đóng băng (*freeze*) trọng số của tất cả các lớp tích chập của mô hình đã huấn luyện sẵn và chỉ huấn luyện lại lớp phân loại (FC layer) cuối cùng cho nhiệm vụ mới. Phù hợp khi bộ dữ liệu mới nhỏ.
  2. **Tinh chỉnh (Fine-Tuning):** Huấn luyện lại các lớp sâu hơn (hoặc toàn bộ mạng) với tốc độ học tập (*learning rate*) rất nhỏ. Phù hợp khi bộ dữ liệu mới đủ lớn và tương tự với dữ liệu huấn luyện gốc.

3 OBJECT DETECTION (OD) - PHÁT HIỆN VẬT THỂ

Phát hiện vật thể bao gồm xác định vật thể (*phân loại*) và *vị trí* bằng hộp giới hạn (bounding box).

3.1 Phát hiện hai giai đoạn (Two-Stage Detectors)

Các mô hình này hoạt động theo hai bước: tạo đề xuất vùng và sau đó phân loại/tinh chỉnh. Độ chính xác cao, tốc độ chậm.

- **Faster R-CNN:** Giai đoạn 1 sử dụng **Region Proposal Network (RPN)** để quét ảnh và đề xuất các vùng có khả năng chứa vật thể (Region of Interest - RoI). Giai đoạn 2 sử dụng các RoI này để trích xuất đặc trưng và thực hiện phân loại, hồi quy hộp giới hạn.

3.2 Phát hiện một giai đoạn (One-Stage Detectors)

Các mô hình này dự đoán hộp giới hạn và lớp trực tiếp, ưu tiên tốc độ, cho phép ứng dụng thời gian thực.

- **YOLO (You Only Look Once):** Chia ảnh thành một lưới $S \times S$. Mỗi ô lưới chịu trách nhiệm dự đoán các hộp giới hạn, điểm tự tin (*confidence score*) và xác suất lớp nếu tâm vật thể rơi vào ô đó.
- **SSD (Single Shot MultiBox Detector):** Sử dụng nhiều bản đồ đặc trưng có kích thước khác nhau (multi-scale feature maps) để phát hiện các vật thể ở các tỉ lệ khác nhau (vật thể nhỏ được phát hiện ở lớp sâu hơn, vật thể lớn ở lớp nông hơn).

3.3 Khái niệm cốt lõi

- **Hộp Neo (Anchor Boxes):** Các hộp giới hạn có kích thước và tỉ lệ cố định được định nghĩa trước. Mô hình học cách điều chỉnh (hồi quy) các hộp neo này để khớp với vật thể thực tế.
- **Intersection over Union (IoU):** Metric đo lường độ trùng lặp giữa hộp dự đoán (B_p) và hộp thực tế (B_{gt}), dùng để đánh giá độ chính xác vị trí. Ngưỡng IoU quyết định một dự đoán có được coi là đúng hay không. Công thức:

$$\text{IoU} = \frac{\text{Diện tích giao nhau}(B_p \cap B_{gt})}{\text{Diện tích hợp lại}(B_p \cup B_{gt})}$$

- **Non-Maximum Suppression (NMS):** Kỹ thuật lọc bỏ các hộp giới hạn trùng lặp cao, chỉ giữ lại hộp có điểm tự tin cao nhất.

4 IMAGE SEGMENTATION (SEG) - PHÂN ĐOẠN ẢNH

Nhiệm vụ phân loại ở cấp độ pixel.

4.1 Phân đoạn Ngữ nghĩa (Semantic Segmentation)

Gán nhãn lớp cho **mỗi pixel** trong ảnh. Tất cả các cá thể cùng lớp được xem là một.

- **Kiến trúc Encoder-Decoder (ví dụ: FCN, SegNet):**
  1. **Encoder (Mã hóa):** Sử dụng các lớp CNN thông thường để trích xuất đặc trưng và giảm kích thước không gian.
  2. **Decoder (Giải mã):** Sử dụng Tích chập Chuyển vị (*Transposed Convolution* hay *Deconvolution*) hoặc *Upsampling* để khôi phục lại kích thước ảnh ban đầu.
- **U-Net:** Kiến trúc chữ U, nổi bật nhờ việc sử dụng **kết nối bỏ qua (skip connections)** giữa Encoder và Decoder. Kết nối này cho phép truyền các đặc trưng chi tiết không gian (từ lớp nông) trực tiếp đến Decoder, giúp cải thiện độ chính xác ranh giới vật thể.

4.2 Phân đoạn Cá thể (Instance Segmentation)

Phát hiện và phân đoạn **từng cá thể** vật thể, ngay cả khi chúng cùng lớp và chồng lấn.

- **Mask R-CNN:** Kiến trúc tiêu chuẩn, mở rộng Faster R-CNN bằng cách thêm một nhánh thứ ba hoạt động song song với nhánh phân loại và hồi quy hộp giới hạn. Nhánh này dự đoán mặt nạ nhị phân (*mask*) cho từng vùng RoI.

5 TRICKS & TIPS CHO THỊ GIÁC MÁY TÍNH SÂU

5.1 Tăng cường Dữ liệu (Data Augmentation)

Là phương pháp tạo ra dữ liệu huấn luyện mới từ dữ liệu hiện có bằng cách áp dụng các biến đổi.

- **Mục đích:** Tăng tính tổng quát hóa của mô hình và chống quá khớp (*overfitting*).
- **Các kỹ thuật cơ bản:** Lật ngang/dọc, Xoay (Rotation), Cắt ngẫu nhiên (Random cropping), Biến đổi màu sắc (Brightness/Contrast/Saturation).
- **Các kỹ thuật nâng cao (Regularization mạnh mẽ):**
  1. **CutOut:** Xóa ngẫu nhiên một vùng hình vuông của ảnh (giống như bị che khuất).
  2. **CutMix/MixUp:** Trộn lẫn hai ảnh và nhãn của chúng một cách tuyến tính hoặc ngẫu nhiên. Ví dụ: trộn 70% ảnh A và 30% ảnh B, nhãn kết quả là 70% nhãn A và 30% nhãn B.

5.2 Chuẩn hóa Hàng loạt (Batch Normalization - BN)

Chèn BN sau lớp tích chập và trước lớp kích hoạt.

- **Động lực:** Giải quyết **Internal Covariate Shift** – hiện tượng phân phối đầu vào của các lớp ẩn thay đổi trong quá trình huấn luyện, làm chậm quá trình hội tụ.
- **Cơ chế:** Chuẩn hóa đầu vào của lớp về mean = 0 và std = 1 cho mỗi batch, sau đó áp dụng phép biến đổi học được ($\gamma x + \beta$) để khôi phục khả năng biểu diễn.
- **Lợi ích:** Tăng tốc độ hội tụ (có thể sử dụng *Learning Rate* cao hơn), giảm sự nhạy cảm với khởi tạo, và đóng vai trò như một hình thức *regularization* nhẹ.

5.3 Tối ưu hóa Tốc độ Học (Learning Rate Scheduling)

Quản lý tốc độ học là rất quan trọng để tránh hội tụ về cực tiểu địa phương xấu.

- **Giảm dần (Decay):** Giảm Learning Rate sau một số epoch cố định (Step Decay) hoặc liên tục (Cosine Annealing).
- **Warm-up:** Bắt đầu với Learning Rate rất nhỏ và tăng dần trong vài epoch đầu tiên. Điều này giúp ổn định các trọng số sau khi khởi tạo và tránh gradient bùng nổ.
- **Cyclical Learning Rates:** Thay đổi Learning Rate giữa giá trị tối thiểu và tối đa theo chu kỳ.

5.4 Regularization

- **Dropout:** Trong mỗi bước huấn luyện, ngẫu nhiên đặt giá trị đầu ra của một phần trăm nơ-ron thành 0. Điều này buộc mạng phải học các biểu diễn mạnh mẽ hơn, không phụ thuộc vào một nhóm nơ-ron cụ thể.
- **L2 Regularization (Weight Decay):** Thêm một số hạng vào hàm mất mát (Loss Function) để phạt các trọng số lớn. Công thức (ví dụ): $L_{\text{total}} = L_{\text{data}} + \frac{\lambda}{2} \sum w^2$.

5.5 Khởi tạo Trọng số Tốt

Sử dụng các phương pháp khởi tạo (**He initialization** cho ReLU, **Xavier initialization** cho Sigmoid/Tanh) để đảm bảo độ lệch chuẩn của các đầu ra của các lớp được giữ ổn định, ngăn chặn sự biến mất hoặc bùng nổ của gradient.

5.6 Ensemble (Hợp nhất)

Huấn luyện nhiều mô hình độc lập (cùng hoặc khác kiến trúc) và lấy kết quả dự đoán trung bình. Kỹ thuật này gần như luôn cải thiện độ chính xác nhưng tăng chi phí tính toán đáng kể.