**Lý thuyết về CNN (Convolutional Neural Network)**

**3.1. Cơ sở lý thuyết**

**3.1.1. Giới thiệu CNN**

* CNN (Convolutional Neural Network) là một loại mạng nơ-ron nhân tạo (ANN) thuộc Deep Learning, được thiết kế đặc biệt để xử lý dữ liệu có cấu trúc dạng lưới, chẳng hạn như hình ảnh hoặc dữ liệu không gian.
* Lấy cảm hứng từ quá trình xử lý thị giác của động vật, CNN mô phỏng cách não bộ nhận diện các đặc trưng không gian thông qua các bộ lọc.
* CNN khai thác hiệu quả cấu trúc không gian của ảnh, cho phép trích xuất đặc trưng tự động mà không cần can thiệp thủ công.
* Hiệu quả vượt trội trong các tác vụ liên quan đến hình ảnh:
  + Phân loại ảnh (image classification).
  + Nhận dạng đối tượng (object detection).
  + Phân đoạn ảnh (image segmentation).
  + Phân tích video (video analysis).
* CNN giảm thiểu số lượng tham số cần huấn luyện so với mạng nơ-ron truyền thống (fully connected) nhờ cơ chế chia sẻ trọng số (weight sharing) trong các bộ lọc.

**3.1.2. Kiến trúc của CNN**

* Một mạng CNN điển hình bao gồm các lớp chính sau:
  + **Lớp tích chập (Convolutional Layer)**: Lớp cốt lõi, trích xuất đặc trưng từ ảnh đầu vào.
    - **Nguyên lý hoạt động**:
      * **Bộ lọc (Filter/Kernel)**: Ma trận nhỏ chứa các trọng số (weights) được học trong quá trình huấn luyện. Bộ lọc trượt (convolve) trên ảnh đầu vào hoặc feature map của lớp trước, thực hiện phép tính vô hướng tại mỗi vị trí.
      * **Feature Map**: Kết quả của phép tích chập, thể hiện mức độ hiện diện của các đặc trưng mà filter đã học. Mỗi filter tạo ra một feature map riêng.
    - **Các tham số quan trọng**:
      * **Filters**: Số lượng filter, càng nhiều filter thì mô hình càng học được nhiều đặc trưng khác nhau.
      * **Kernel\_size**: Kích thước của filter (ví dụ: 3x3, 5x5).
      * **Strides**: Bước nhảy của filter khi trượt trên ảnh, ảnh hưởng đến kích thước feature map.
      * **Padding**:
        + "Valid" (no padding): Không thêm pixel, feature map đầu ra nhỏ hơn.
        + "Same": Thêm pixel (thường là 0) để feature map đầu ra có kích thước giống đầu vào.
      * **Activation**: Hàm kích hoạt áp dụng sau phép tích chập (thường là ReLU để tăng tính phi tuyến).
      * **Input\_shape**: Kích thước ảnh đầu vào (ví dụ: (28, 28, 1) cho ảnh xám 28x28 pixel), chỉ áp dụng cho lớp đầu tiên.
  + **Lớp gộp (Pooling Layer)**: Giảm kích thước feature map, tập trung vào các đặc trưng quan trọng, đồng thời giảm nguy cơ overfitting.
    - **Nguyên lý hoạt động**:
      * **Vùng gộp**: Một vùng nhỏ (ví dụ: 2x2) trên feature map được dùng để gộp.
      * **Max Pooling**: Lấy giá trị lớn nhất trong vùng gộp.
      * **Average Pooling**: Tính giá trị trung bình trong vùng gộp.
      * **Stride**: Bước nhảy của vùng gộp, stride lớn giảm kích thước feature map nhiều hơn.
    - **Các tham số quan trọng**:
      * **Pool\_size**: Kích thước vùng gộp (ví dụ: (2, 2)).
      * **Strides**: Bước nhảy của vùng gộp, thường bằng pool\_size.
  + **Lớp làm phẳng (Flatten Layer)**: Chuyển feature map đa chiều thành vector một chiều để chuẩn bị cho lớp fully connected.
    - **Nguyên lý hoạt động**: Ví dụ, chuyển ma trận 5x5x64 thành vector 1 chiều (1600 phần tử).
  + **Lớp kết nối đầy đủ (Fully Connected Layer)**: Lớp cuối cùng, thực hiện phân loại hoặc hồi quy dựa trên các đặc trưng đã trích xuất.
    - **Nguyên lý hoạt động**:
      * Mỗi nơ-ron thực hiện phép toán tuyến tính với đầu vào từ lớp trước, sau đó áp dụng hàm kích hoạt.
      * Trọng số được học để xác định mối quan hệ giữa các đặc trưng.
    - **Tham số quan trọng**:
      * **Units**: Số lượng nơ-ron trong lớp.
      * **Activation**: Hàm kích hoạt (ví dụ: softmax cho phân loại đa lớp).
  + **Hàm kích hoạt**:
    - **ReLU (Rectified Linear Unit)**: , tăng tính phi tuyến, tránh gradient biến mất.
    - **ELU (Exponential Linear Unit)**: nếu , hoặc nếu , cải thiện tốc độ học.
    - **Softmax**: Chuyển đầu ra thành xác suất cho phân loại đa lớp,

**3.1.3. Quá trình huấn luyện CNN**

* **Feedforward**:
  + Dữ liệu đầu vào đi qua các lớp của CNN (tích chập → gộp → làm phẳng → fully connected).
  + Feature map và giá trị đầu ra được tính toán tại mỗi lớp.
* **Hàm mất mát (Loss Function)**:
  + Đo lường sự sai khác giữa dự đoán và nhãn thực tế.
  + **Categorical Crossentropy** (cho phân loại đa lớp, như MNIST):
    - ​: One-hot encoding của nhãn thực tế.
    - ​: Phân phối xác suất dự đoán.
* **Backpropagation**:
  + Thuật toán lan truyền ngược, tính đạo hàm của hàm mất mát theo các tham số (trọng số, bias).
  + Sử dụng quy tắc chuỗi (chain rule) để tính gradient hiệu quả.
* **Thuật toán tối ưu (Optimizer)**:
  + **Mini Batch Gradient Descent**: Cập nhật trọng số dựa trên gradient của một lô nhỏ (mini-batch).
  + **Adam (Adaptive Moment Estimation)**: Kết hợp momentum và RMSProp, điều chỉnh tốc độ học thích nghi.
  + **RMSProp**: Tối ưu dựa trên trung bình động của gradient bình phương, phù hợp với dữ liệu không ổn định.

**3.1.4. Ứng dụng, ưu và nhược điểm của CNN**

* **Ứng dụng**:
  + **Phân loại hình ảnh**: Phân loại ảnh thành các danh mục (ví dụ: mèo/chó, ô tô/xe tải, hoa/động vật).
  + **Phát hiện đối tượng**: Xác định và định vị đối tượng trong ảnh (ví dụ: người, ô tô, tòa nhà).
  + **Phân đoạn hình ảnh**: Gắn nhãn từng pixel để phân đoạn đối tượng, hữu ích trong y tế và robot.
  + **Phân tích video**: Theo dõi đối tượng, phát hiện sự kiện (ví dụ: giám sát video, giao thông).
* **Ưu điểm**:
  + Đạt độ chính xác cao trong nhận diện hình ảnh (phân loại, phát hiện, phân đoạn).
  + Hiệu quả cao khi triển khai trên GPU nhờ tính toán song song.
  + Học được các đặc trưng tổng quát, ít bị ảnh hưởng bởi nhiễu và thay đổi dữ liệu đầu vào.
  + Có thể áp dụng cho dữ liệu chưa từng thấy trong huấn luyện.
  + Dễ dàng điều chỉnh kiến trúc để thực hiện nhiều nhiệm vụ khác nhau.
* **Nhược điểm**:
  + Yêu cầu lượng dữ liệu huấn luyện lớn để đạt hiệu suất tốt, dễ gây overfitting nếu dữ liệu ít hoặc không đa dạng.
  + Đòi hỏi nhiều tài nguyên tính toán (CPU/GPU), thời gian huấn luyện có thể lâu.
  + Khó diễn giải (black box), khó hiểu tại sao mô hình đưa ra dự đoán cụ thể.

**3.2. Áp dụng CNN vào bài toán phân loại**

**3.2.1. CNN trong bài toán phân loại số viết tay (MNIST)**

* **Đặc điểm của MNIST**:
  + Bộ dữ liệu ảnh chữ số viết tay (0-9), mỗi ảnh 28x28 pixel, thang xám.
  + Chữ số có các đặc trưng đa dạng: đường cong, đường thẳng, góc cạnh.
  + Trích xuất đặc trưng thủ công rất khó khăn và tốn thời gian.
* **Tính chất của MNIST**:
  + Dữ liệu là ảnh, có cấu trúc không gian, phù hợp với CNN.
* **Lý do chọn CNN**:
  + **Trích xuất đặc trưng tự động**: CNN tự động học và trích xuất các đặc trưng quan trọng từ ảnh (đường thẳng, góc cạnh, đường cong) mà không cần can thiệp thủ công.
  + **Hiệu quả không gian**: Phép tích chập và gộp tận dụng thông tin không gian của ảnh để dự đoán chính xác.
  + **Hiệu suất cao**: CNN đạt độ chính xác vượt trội trên MNIST (~98-99%), cao hơn các thuật toán học máy truyền thống.
  + **Phức tạp phù hợp**: Dù MNIST là bài toán đơn giản, vẫn cần mô hình phức tạp để đạt kết quả tốt, CNN là lựa chọn hợp lý.
  + **Bộ lọc học được**: Các bộ lọc trong lớp tích chập tự động phát hiện các mẫu quan trọng (edges, curves) mà không cần thiết kế thủ công.
* **So sánh**:
  + CNN thường cho độ chính xác cao hơn so với các thuật toán học máy truyền thống (như SVM, k-NN) trên bài toán MNIST.