**CNN**

**I/**

**1. Bối cảnh hình thành CNN**

* **Trước LeNet-5**:
  + Ý tưởng về mạng nơ-ron nhiều lớp đã xuất hiện từ thập niên 1980, nhưng việc huấn luyện các mô hình sâu gặp khó khăn do giới hạn tính toán và vấn đề gradient biến mất.
  + Sự phát triển của thuật toán Backpropagation vào cuối những năm 1980 và 1990, cùng với khả năng xử lý của máy tính, đã đặt nền móng cho CNN.
* **LeNet-5 (1998)**:
  + Được Yann LeCun phát triển để phân loại chữ viết tay (như mã bưu chính trên séc).
  + **Cấu trúc chính**: Gồm 7 lớp (không kể lớp input), kết hợp các lớp tích chập, pooling và fully connected.
  + **Ứng dụng**: Thành công đầu tiên trong thị giác máy tính, mở đường cho các ứng dụng CNN hiện đại.

**2. Các cột mốc quan trọng khác**

* **2006**: Geoffrey Hinton giới thiệu **Deep Belief Networks (DBN)**, mở đường cho sự hồi sinh của mạng học sâu.
* **2009**: ImageNet dataset ra đời, cung cấp một kho dữ liệu hình ảnh lớn giúp thúc đẩy phát triển các mô hình học sâu.
* **2012**: AlexNet giành chiến thắng trong cuộc thi ImageNet với tỷ lệ lỗi giảm 10% so với phương pháp tốt nhất trước đó, khẳng định sức mạnh của CNN.
* **2014**: Sự xuất hiện của **VGGNet** và **GoogleNet** đã cải thiện hiệu quả và độ sâu của CNN.
* **2015**: **ResNet (Residual Networks)** được giới thiệu bởi Microsoft, mở ra khả năng huấn luyện các mạng cực kỳ sâu (hàng trăm lớp) mà không bị mất gradient.

**3. Ưu điểm và nhược điểm của CNN**

**Ưu điểm:**

1. **Học tự động các đặc trưng**: CNN tự động tìm ra các đặc trưng phù hợp nhất từ dữ liệu mà không cần can thiệp thủ công.
2. Nhờ chia sẻ tham số và xử lý cục bộ thông qua tích chập, các mô hình CNN hiệu quả hơn về mặt tính toán so với các mô hình khác như FNN hoặc MLP.
3. **Ứng dụng rộng rãi**: Có thể áp dụng cho dữ liệu hình ảnh, âm thanh, video, và thậm chí là văn bản.

**Nhược điểm:**

1. **Yêu cầu dữ liệu lớn**: CNN cần tập dữ liệu lớn để hoạt động tốt.
2. **Thời gian huấn luyện lâu**: Việc huấn luyện một CNN sâu thường đòi hỏi tài nguyên tính toán cao.
3. **Khả năng giải thích hạn chế**: Khó hiểu được logic đằng sau các quyết định của CNN.

**4. Ứng dụng của CNN trong đời sống hiện đại**

1. **Thị giác máy tính (Computer Vision)**:
   * Nhận dạng khuôn mặt (Face Recognition).
   * Phát hiện vật thể (Object Detection).
   * Phân loại hình ảnh (Image Classification).
2. **Y tế**:
   * Chẩn đoán hình ảnh y khoa như X-quang, MRI, hoặc CT-scan.
   * Phân đoạn khối u trong ảnh y tế.
3. **Tự động hóa và xe tự lái**:
   * Nhận diện làn đường, biển báo giao thông.
   * Dự đoán chuyển động của các đối tượng trên đường.
4. **Xử lý ngôn ngữ tự nhiên (NLP)**:
   * Mô hình hóa chuỗi văn bản bằng CNN để phân loại văn bản, phát hiện spam.
5. **Ứng dụng thương mại**:
   * Lọc nội dung không phù hợp trên mạng xã hội.
   * Gợi ý sản phẩm trên các nền tảng thương mại điện tử.

**5. Các cải tiến quan trọng của CNN**

* **Dropout (2012)**: Được giới thiệu trong AlexNet để giảm hiện tượng overfitting.
* **Batch Normalization (2015)**: Giúp tăng tốc độ huấn luyện và cải thiện hiệu năng của mạng.
* **Residual Connections (ResNet, 2015)**: Giải quyết vấn đề gradient biến mất trong các mạng sâu.
* **Attention Mechanism (2017)**: Tăng cường khả năng tập trung của CNN vào các vùng quan trọng trong hình ảnh.

Convolution neural network gồm những lớp cơ bản sau:

1. **Convolutional layer**

Đây chính là lớp đóng vai trò mấu chốt của CNN, khi layer này đảm nhiệm việc thực hiện mọi tính toán. Stride, padding, filter map, feature map là những yếu tố quan trọng nhất của convolutional layer.

* Cơ chế của CNN là tạo ra các filter áp dụng vào từng vùng hình ảnh. Các filter map này được gọi là ma trận 3 chiều, bên trong chứa các parameter dưới dạng những con số.
* Stride là sự dịch chuyển filter map theo pixel dựa trên giá trị từ trái sang phải.
* **Padding**: Là các giá trị 0 được thêm cùng lớp input.
* **Feature map:** Sau mỗi lần quét, một quá trình tính toán sẽ được thực hiện. Feature map sẽ thể hiện kết quả sau mỗi lần filter map quét qua input.

1. **Relu layer**

Còn có tên gọi khác là activation function, đây là một hàm được kích hoạt trong neural network. Có tác dụng mô phỏng các neuron có tỷ lệ truyền xung qua axon. Trong activation function chúng còn có hàm nghĩa là: Relu, Tanh, Sigmoid, Maxout, Leaky,… Relu layer được ứng dụng phổ biến trong việc huấn luyện nơ-ron do sở hữu nhiều ưu điểm tiên tiến.

1. **Pooling layer**

Khi nhận phải đầu vào quá lớn, các lớp pooling layer sẽ được xếp giữa những lớp Convolutional layer nhằm mục đích giảm parameter. Pooling layer được chia thành 2 loại phổ biến là max pooling và average.

1. **Fully connected layer**

Khi 2 lớp convolutional layer và pooling layer nhận được ảnh truyền, lớp này sẽ có nhiệm vụ xuất kết quả. Khi ta nhận được kết quả là model đọc được thông tin ảnh, ta cần phải tạo sự liên kết để cho ra nhiều output hơn. Đây chính là lúc các lập trình viên sử dụng fully connected layer. Hơn nữa, nếu fully connected layer có dữ liệu về hình ảnh thì chúng sẽ chuyển thành mục chưa được phân chia chất lượng.

1. **· Activtation function:**

Thông thường sẽ sử dụng Sigmoid hoặc Tanh

Output của lớp fully connected cuối cùng đi qua hàm softmax

**II/**

Một cấu trúc cơ bản nhất của CNN sẽ bao gồm 3 phần chủ yếu, đó là:

* **Local receptive field (trường cục bộ):**Nhiệm vụ của trường cục bộ là phân tách và lọc dữ liệu cũng như thông tin ảnh, sau đó chọn ra các vùng ảnh có giá trị sử dụng cao nhất.
* **Shared weights and bias (trọng số chia sẻ):**Trong mạng CNN, thành phần này có tác dụng giảm thiểu tối đa lượng tham số có tác dụng lớn. Trong mỗi convolution sẽ chứa nhiều feature map khác nhau, mỗi feature lại có khả năng giúp nhận diện một số feature trong ảnh.
* **Pooling layer (lớp tổng hợp):**Pooling layer là lớp cuối cùng, với khả năng đơn giản hóa thông tin đầu ra. Khi đã hoàn tất tính toán và quét qua các lớp, pooling layer sẽ được tạo ra nhằm mục đích lược bớt các thông tin không cần thiết và tối ưu đầu ra. Điều này giúp người dùng nhận được kết quả ưng ý và đúng với yêu cầu hay mong muốn.

**III/**

Nhằm lựa chọn được tham số phù hợp nhất cho CNN, bạn nên lưu ý đến số lượng các yếu tố sau:

* **Convolution layer:**Nếu lớp này có số lượng lớn hơn,chương trình chạy của bạn sẽ càng được cải thiện và tiến bộ. Sử dụng layer với số lượng nhiều có thể giúp các tác động được giảm một cách đáng kể. Trong đa phần các trường hợp, chỉ cần khoảng 3 đến 5 lớp là bạn sẽ thu về kết quả như mong đợi.
* **Filter size:**Thông thường, các filter size sẽ có kích thước là 3×3 hoặc 5×5.
* **Pooling size:**Với các hình ảnh thông thường, bạn nên sử dụng loại kích thước 2×2. Nếu đầu vào xuất hiện dạng hình ảnh lớn hơn thì bạn nên chuyển sang dùng loại 4×4.
* **Train test:** Càng thực hiện nhiều lần train test, bạn càng có nhiều khả năng thu được các parameter tốt nhất, giúp mô hình “thông minh” và hiệu quả hơn.

**Sự tích chập trong ảnh màu RGB:**

Trên một lớp tích chập, có thể áp dụng nhiều hạt nhân, trong đó mỗi hạt nhân có trọng số và độ lệch riêng.

Mục đích: Điều này cho phép mạng trích xuất nhiều tính năng khác nhau từ đầu vào.

Đầu ra của một lớp tích chập sau đó được truyền qua một trong hai hàm kích hoạt phổ biến:

Sigmoid: Một hàm kích hoạt phi tuyến tính ánh xạ các giá trị giữa 0 và 1.

Tanh: Một hàm kích hoạt phi tuyến tính khác ánh xạ các giá trị giữa -1 và 1.

**AlexNet**

1. Introduction

- Được giới thiệu bởi Alex Krizhevsky, Ilya Sutskever, và Geoffrey Hinton trong cuộc thi ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012.

- Đánh dấu sự nổi bật của Deep Learning trong lĩnh vực thị giác máy tính.

2. Tại sao AlexNet là một bước đột phá quan trọng?

- Năm 2009, tập dữ liệu ImageNet, chứa 1 triệu hình ảnh được phân bổ đều trên 1.000 lớp, đã được giới thiệu

- Các phương pháp truyền thống như SVM và Rừng ngẫu nhiên không thể khai thác đầy đủ thông tin từ tập dữ liệu ImageNet.

3. Kiến trúc

Convolution Layers: 5 convolutional layer.

Pooling Layers: 3 pooling layer.

Fully Connected Layers: 3 fully connected layer.

Activation Function:

ReLU được áp dụng cho đầu ra của Convolutional Layers và hai Fully Connected layer đầu tiên.

Logits sau FC3 được truyền qua hàm softmax.

Áp dụng Dropout để giảm overfitting.

Sử dụng hai mạng song song để xử lý dữ liệu (để tận dụng khả năng GPU hạn chế tại thời điểm đó).

Sử dụng GPU để huấn luyện giúp tăng hiệu quả.

Phép điều chuẩn dữ liệu (Data Augmentation) như lật ảnh và xoay.

Kết hợp Normalization sau mỗi lớp tích chập (Local Response Normalization - LRN).

4. Tại sao sử dụng ReLU mà khổng phải hàm kích hoạt nào khác?

- ReLU đơn giản, tăng tốc độ tính toán.

- Giảm độ phức tạp của mô hình.

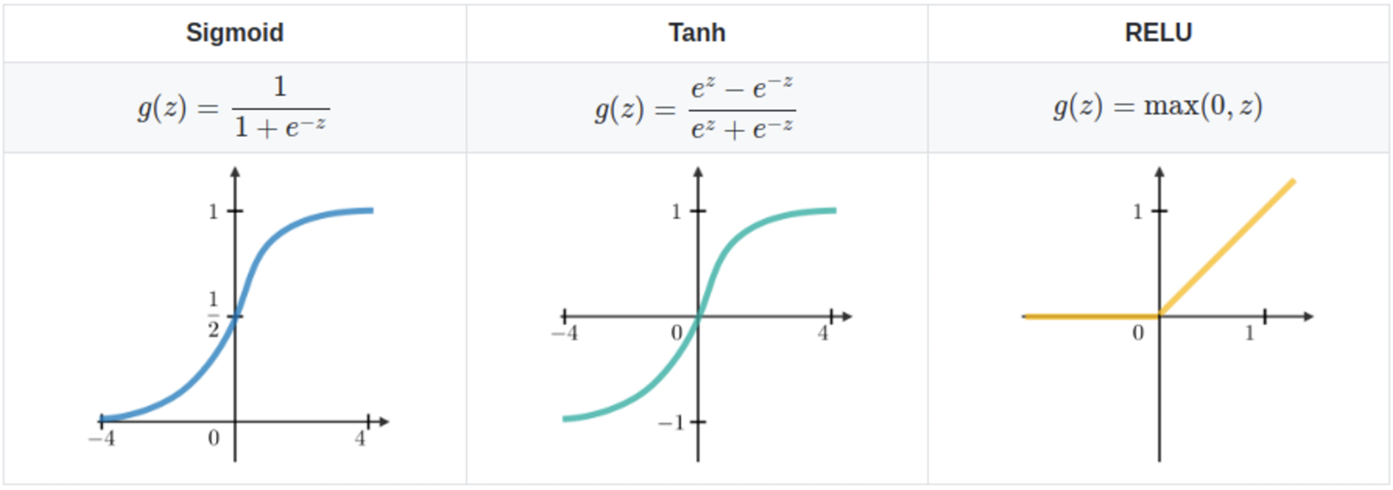
- Tránh được vấn đề biến mất gradient.

5. Nhược điểm của ReLU

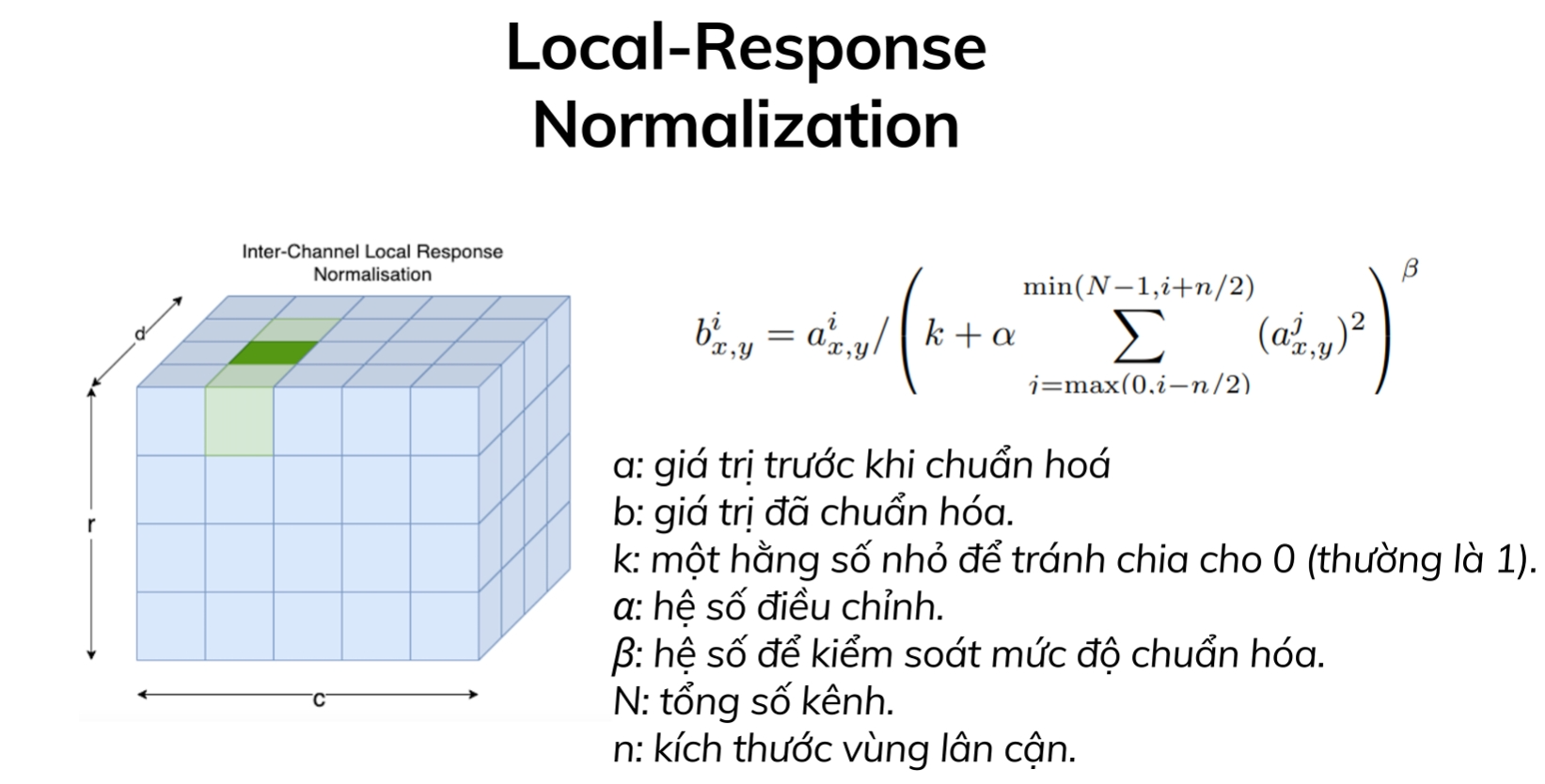
- Các giá trị lớn hơn 0 không thay đổi.

- Các giá trị nhỏ hơn hoặc bằng 0 được đặt thành 0. => Điều này có thể dẫn đến bùng nổ gradient hoặc mất thông tin trong quá trình học.

6. Hàm kích hoạt: Sigmoid, Tanh, amd ReLU



7. Local Response Normalization



8. So sánh mô hình AlexNet với các mô hình trước đây và sau này

**So với các mô hình ra mắt trước**

* Cho phép sử dụng GPU để huấn luyện.
* Đạt được tỷ lệ lỗi thấp hơn đáng kể so với các mô hình trước đó.
* Đóng vai trò là kiến trúc nền tảng cho các mô hình sau này như ResNet và GoogleNet.

**So với các mô hình ra mắt sau**

* Mất nhiều thời gian huấn luyện do kích thước mô hình lớn (60 triệu tham số).
* Kích thước lớn khiến mô hình không phù hợp với các thiết bị có phần cứng hạn chế.
* Cấu trúc tương đối đơn giản so với các mô hình hiện đại hơn.