TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN MÔN HỌC

CHUYÊN ĐỀ THIẾT KẾ HỆ VI MẠCH I

MẠNG NƠ RON TÍCH CHẬP CNN

ALEXNET

- **Giảng viên hướng dẫn: ThS.Trương Văn Cương.**
- **Sinh viên thực hiện:**
  1. Võ Phúc Vinh Khang (17520621)
  2. Lê Nguyễn Anh Huy (17520572)

MỤC LỤC:

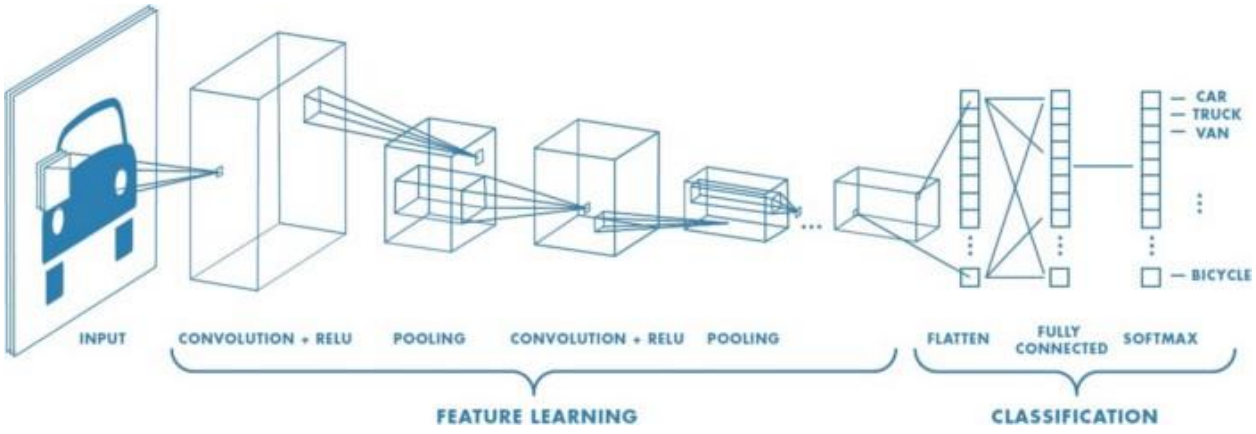
I.Tổng quan

1. Tổng quan mạng CNN:

_ Convolutional Neural Network(CNN) hay mạng nơ ron tích chập là một mô hình deep learning tiên tiến giúp chúng ta xây dựng những hệ thống thông minh với độ chính xác cao. CNN thường được sử dụng trong các bài toán nhận dạng vật thể trong ảnh.

2. Cấu trúc mạng CNN

_ Mô hình CNN là tập hợp các layer liên kết với nhau thông qua cơ chế Convolution. Layer tiếp theo là kết quả Convolution từ layer trước đó, nhờ vậy mà ta có được các kết nối cục bộ. Như vậy mỗi neuron ở lớp kế tiếp sinh ra từ kết quả của filter áp đặt lên một vùng ảnh cục bộ của neuron trước đó.



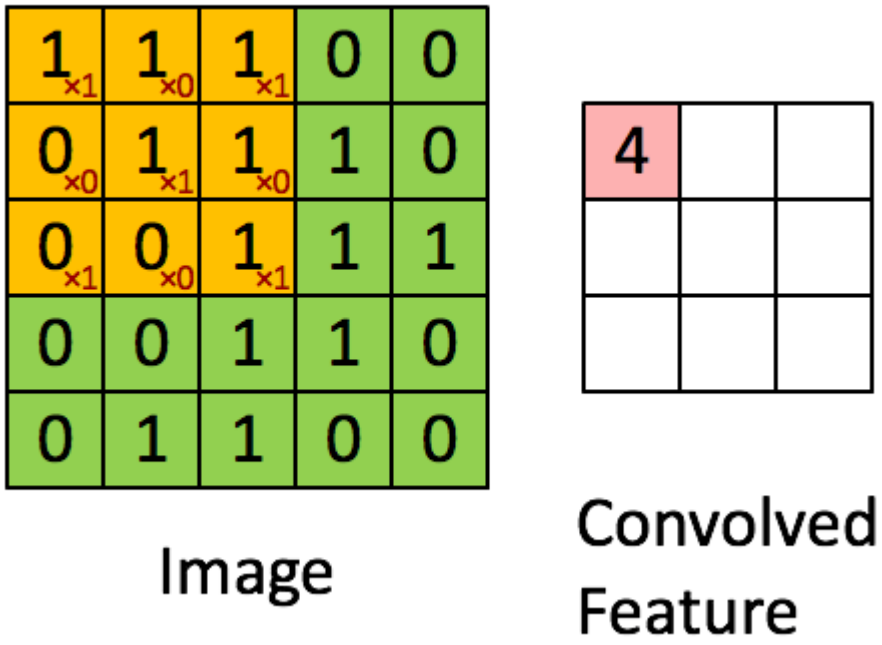
Mô hình tổng quan của một mạng CNN

_ Mỗi một lớp sử dụng 1 lớp filter khác nhau và thông thường có rất nhiều lớp filter như vậy và kết hợp kết quả của chúng lại. Các layer thường được sử dụng trong hệ thống như pooling hay subsampling layer dùng để chắt lọc các thông tin hữu ích.

_ Thông số của các filter sẽ được tự động học trong quá trình training và dựa trên cách thức training. Ví dụ trong tác vụ phân lớp ảnh, CNNs sẽ cố gắng tìm ra thông số tối ưu cho các filter tương ứng theo thứ tự raw pixel => edges => shapes => facial => high-level features. Layer cuối cùng được dùng để phân lớp ảnh.

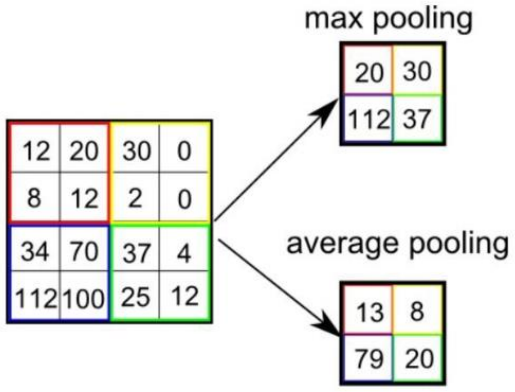
3. Các thành phần trong mạng CNN.

- *Convolutional layer* : hay còn gọi là các filter được dùng để lấy các thông tin trong các bức ảnh. Trong một Convolution layer thường có nhiều kernel có cùng kích thước. Thông thường thì width và height của 1 kernel bằng nhau và depth thường bằng số lượng kênh màu.



Thuật toán convolution.

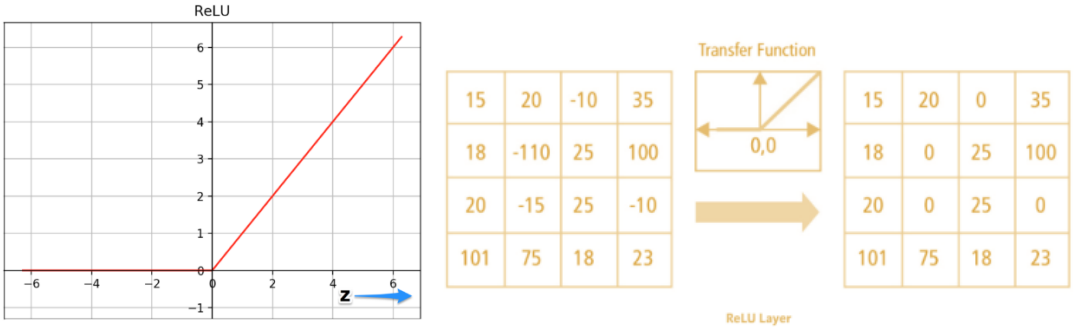
- *Pooling* : thường được sử dụng để giảm chiều rộng và chiều dài của 1 bức ảnh nhưng vẫn giữ nguyên chiều sâu. Overlapping maxpooling cũng giống như maxpooling ngoại trừ việc là một window của bước này sẽ có 1 phần chồng lên window của bước tiếp theo.



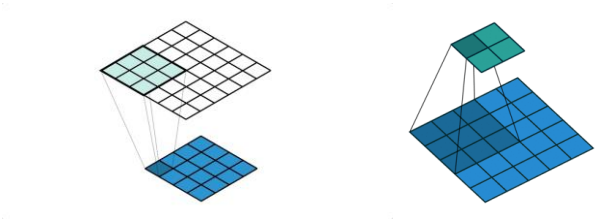
Ví dụ về pooling với stride = 2

+ Có 2 loại pooling : max pooling và average pooling. Max pooling sẽ trả về giá trị lớn nhất từ phần ảnh được lọc bởi filter. Trong khi đó average pooling sẽ trả về giá trị trung bình của tất cả các giá trị trong phần hình ảnh đi qua bộ filter. Các hệ thống CNN hiện nay thông thường sẽ sử dụng maxpooling vì nó có thể loại bỏ các nguồn nhiễu và thực hiện khử nhiễu song song với việc giảm kích thước hình ảnh.

- *Hàm phi tuyến* : trước đây trong các mô hình , người ta thường sử dụng hàm kích hoạt tanh hoặc sigmoid để huấn luyện mô hình neural network. Đến khi AlexNet ra đời, nó được sử dụng hàm ReLU. Người ta chỉ ra rằng khi sử dụng hàm ReLU, mô hình CNN sẽ huấn luyện nhanh hơn so với việc sử dụng hàm sigmoid hoặc tanh. Và 1 điều là đôi khi các giá trị đầu ra sẽ có giá trị âm , việc sử dụng hàm ReLU thay cho các hàm sigmoid hoặc tanh sẽ giúp hình ảnh đầu ra không có giá trị âm vì ReLU sẽ chuyển hóa các giá trị âm thành giá trị 0.



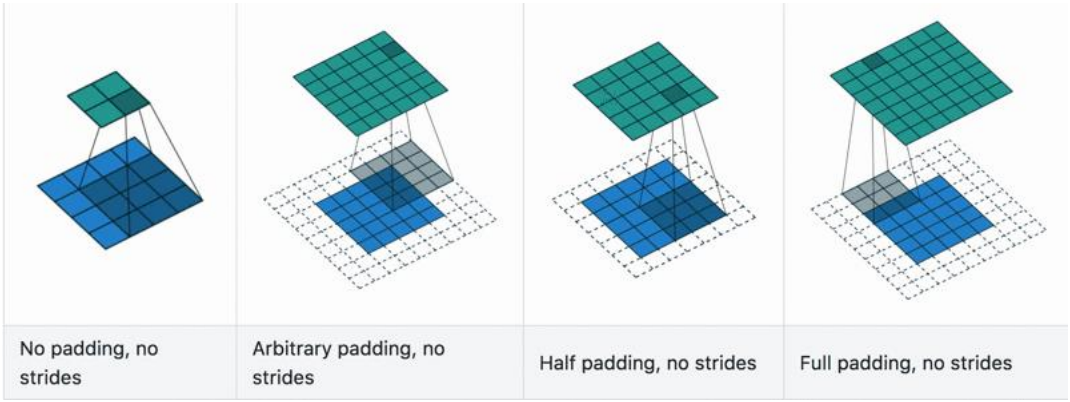
- *Stride(bước nhảy):* là số pixel thay đổi trên ma trận đầu vào. Khi stride là 1 thì ta di chuyển các kernel 1 pixel. Khi stride bằng 2 thì ta di chuyển các kernel đi 2 pixel và tiếp tục như vậy.



Stride 1. Stride 2.

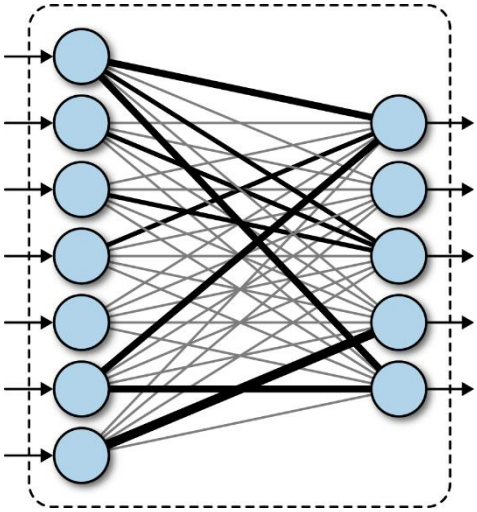
- *Padding(đường viền):* vì hình ảnh sau mỗi lần đi qua lớp filter sẽ nhỏ đi , nhưng đôi khi ta không muốn giảm kích thước hình ảnh sau mỗi lần lọc. Khi đó ta sử dụng padding để hình ảnh sau khi lọc sẽ vẫn giữ nguyên kích thước ban đầu. Padding sẽ có 2 cách sử dụng: chèn thêm các số 0 vào 4 đường biên của hình ảnh hoặc sử dụng giá trị pixel tại mỗi đường

biên. Việc sử dụng padding tại đường biên sẽ không thay đổi quá nhiều giá trị đầu ra của hình ảnh sau khi lọc.



Kỹ thuật padding.

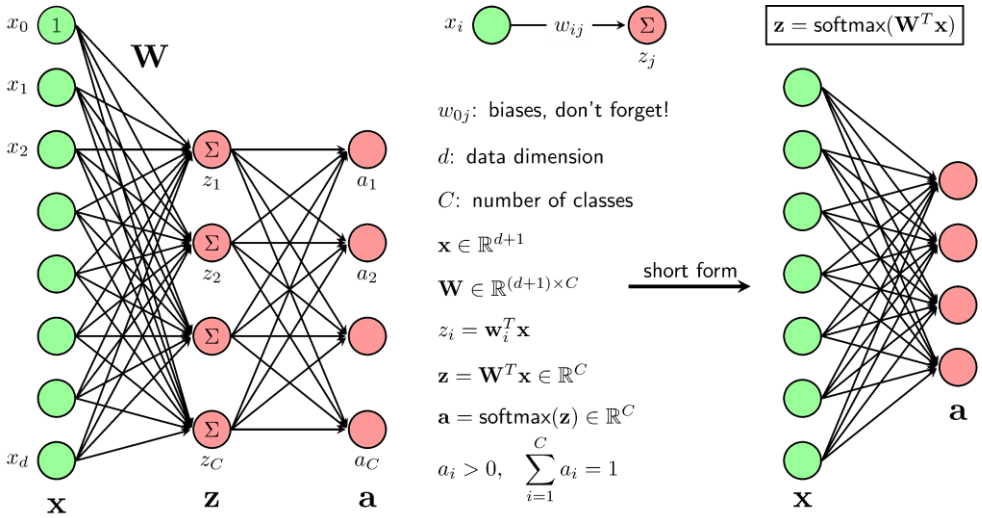
- *Fully connected:* sau khi ảnh đi qua nhiều lớp convolution và pooling thì hệ thống đã được học tương đối các đặc điểm ảnh thì tensor của output của layer cuối cùng được biến thành vector và đưa vào 1 lớp được kết nối như mạng neuron. Với FC layer được kết hợp với các tính năng lại với nhau để tạo thành mô hình. Cuối cùng sử dụng softmax hoặc sigmoid để phân loại đầu ra.



Lớp Fully Connected với 7 node input và 5 node output.

$$y_i = \sum_{k=0}^{W*H-1} x_k * w_k + b_i, 0 \leq i < N$$

+ Softmax: biến vector k chiều có các giá trị thực bất kỳ thành vector k chiều có giá trị thực có tổng bằng 1. Giá trị nhập vào có thể âm , dương , bằng 0 hoặc lớn hơn 1 nhưng hàm softmax sẽ luôn biến chúng thành giá trị nằm trong khoảng (0,1].



Hàm kích hoạt softmax kết nối với hàm Fully Connected

II. Tổng quan về mạng AlexNET

1. Lịch sử hình thành

_ AlexNet là 1 mạng nơ-ron tích chập(CNN) , thiết kế bởi Alex Krizhevsky phối hợp với Sutkever và Geofrey Hinton.

_ Dành chiến thắng trong cuộc thi ImageNet LSVRC-2012 .

_ Đạt lỗi top 5 là 15,3% thấp hơn 10,8 điểm phần trăm so với mạng về nhì.

2. Đặc điểm của Alexnet

_ Sử dụng hàm Relu thay cho sigmiod hoặc tanh để xử lý với non-linearity.

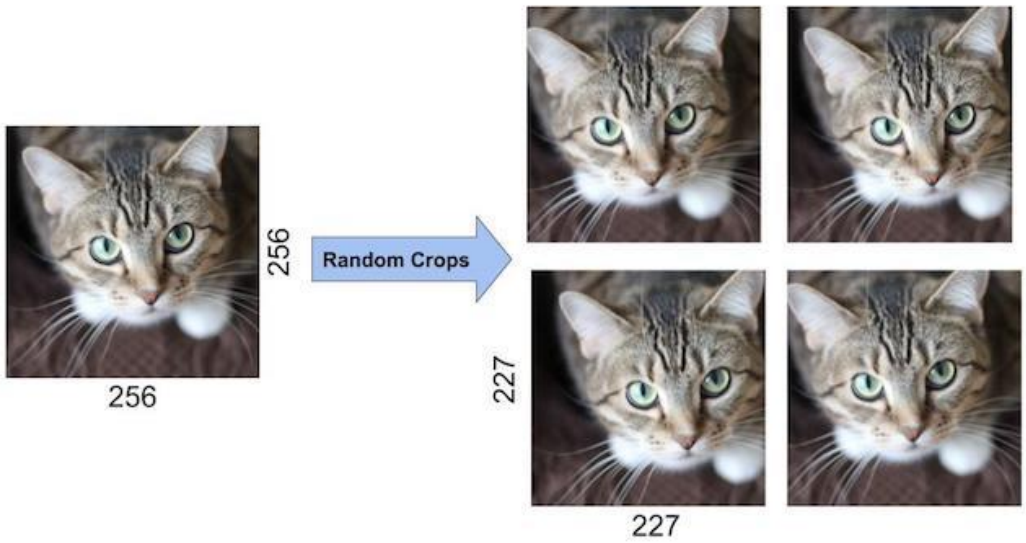
_ Sử dụng dropout như 1 phương pháp regularization , giúp những mô hình tránh việc bị overfitting và giảm thời gian huấn luyện mô hình.

_ Sử dụng overlap pooling để giảm size của mạng.

_ Sử dụng local respone normalization để chuẩn hóa ở mỗi layer.

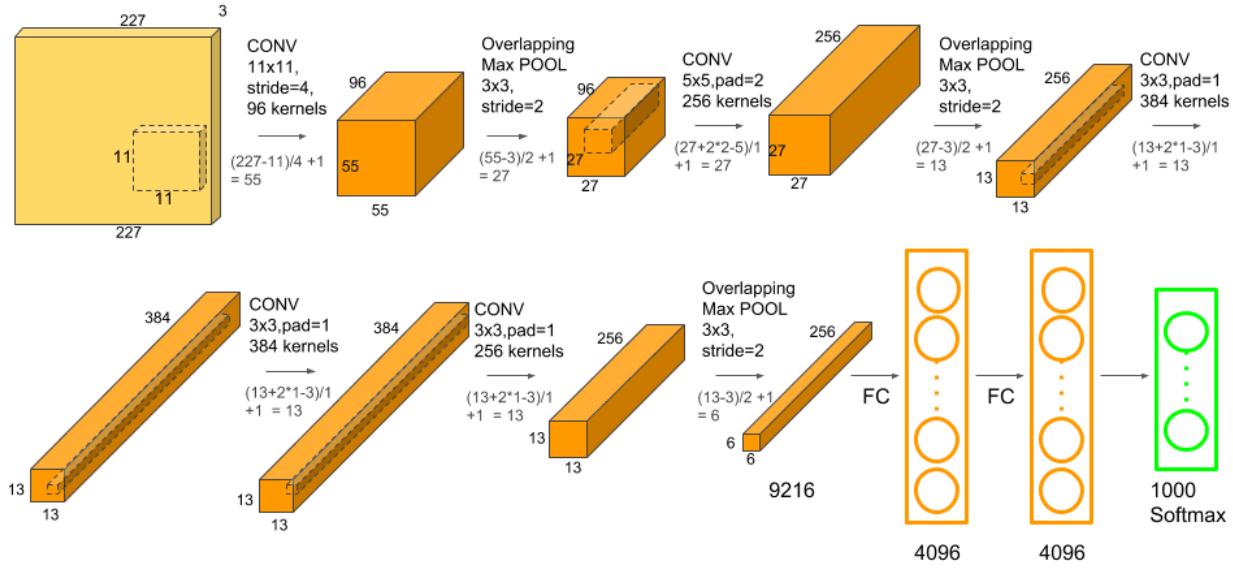
_ Sử dụng kỹ thuật data augmentation để tạo thêm data training bằng cách translations, horizontal reflections.

3. Kiến trúc AlexNet



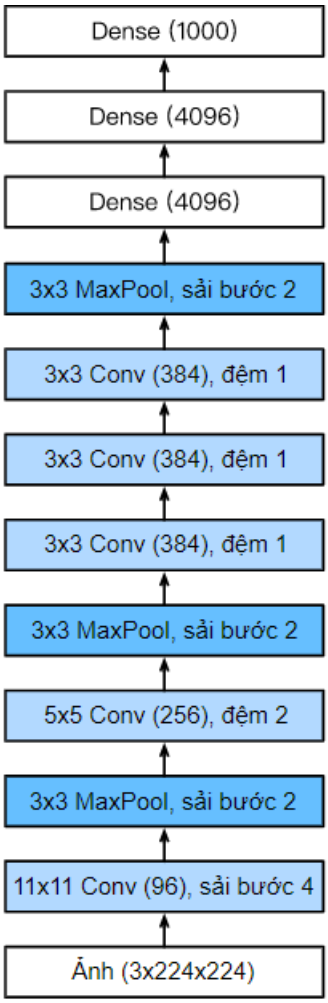
Input ảnh được resize về 256x256 và random crop về 227x227.

_ Gồm 8 lớp bao gồm 5 lớp Convolutional layer và 3 lớp fully connected. Activation ReLU được sử dụng sau mỗi lớp convolution và fully connected. Input đầu vào là 1 bức ảnh với kích thước 227x227x3 với 1000 class khác nhau.



Kiến trúc của mạng AlexNET.

- Layer thứ 1: 96 lớp filter với kích thước 11x11x3 , không sử dụng kĩ thuật padding , bước nhảy = 4, ảnh đầu ra có kích thước là 55x55x96. Max pooling layer : kích thước 3x3 , bước nhảy = 2, activation = ReLU ,output đầu ra là 27x27x96.
- Layer thứ 2: 256 lớp filter với kích thước 3x3x96, bước nhảy = 1 , padding = 0 ,activation= ReLU, output đầu ra có kích thước 27x27x256. Maxpooling layer: kích thước 3x3, bước nhảy = 2, padding = 0 , output có kích thước 13x13x256.
- Layer thứ 3: 384 lớp filter với kích thước 3x3x256, bước nhảy = 1 , padding = 0,activation = ReLu, output có kích thước 13x13x384.
- Layer 4: 384 lớp filter, bước nhảy = 1, padding = 0, activation = ReLU, output có kích thước 13x13x384.
- Layer thứ 5: 256 lớp filter, bước nhảy = 1, padding = 0, activation = ReLU, output có kích thước 13x13x256.
  Pooling layer: kích thước 3x3, bước nhảy = 2, padding = 0, output có kích thước 6x6x256.
- Flatten: kích thước 256x6x6.
- Fully connected layer 1: activation = ReLU, output = 4096 + dropout(0.5).
- Fully connected layer 2: activation = ReLU, output = 4096 + dropout(0.5).
- Fully connected layer 3: activation = softmax, output= 1000 class.



Các layer mô hình AlexNET.

II. Mục tiêu

- Số class thực hiện: 2 class (gấu bông dạng dài và gấu bông dạng tròn)

- Input đầu vào : kiểu integer (từ dữ liệu value_RGB output image của hàm Python)
- Output đầu ra : kiểu integer (từ dữ liệu integer để cho ra output image qua Python)
- Dự tính thực hiện : nhận diện hình dạng gấu bông

IV. Nội dung

1. Tổng quan
2. Hiện thực
3. Đánh giá và kết quả