ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

CHUYÊN ĐỀ
AlexNet

MỤC LỤC

CHƯƠNG 1 CNN ARCHITECTURES

1.1 AlexNet Architecture

1.1.1 Kiến trúc mạng

- Full (simplified) AlexNet architecture:
- [227x227x3] INPUT
- [55x55x96] CONV1: 96 11x11 filters at stride 4, pad 0
- [27x27x96] NORM1: Normalization layer
- [27x27x96] MAX POOL1: 3x3 filter at stride 2
- [27x27x256] CONV2: 256 5x5 filters at stride 1, pad 2
- [13x13x256] NORM2: Normalization layer
- [13x13x256] MAX POOL2: 3x3 filters at stride 2
- [13x13x384] CONV3: 384 3x3 filters at stride 1, pad 1
- [13x13x384] CONV4: 384 3x3 filters at stride 1, pad 1
- [13x13x256] CONV5: 256 3x3 filters at stride 1, pad 1
- [6x6x256] MAX POOL3: 3x3 filters at stride 2
- [4096] FC6: 4096 neurons
- [4096] FC7: 4096 neurons
- [1000] FC8: 1000 neurons (class scores)

1.1.2 INPUT

- INPUT: Ảnh đầu vào kích thước $[227\times227\times3]$.

1.1.3 CONV1

- Input: Ảnh đầu vào $\mathbf{M_1}$ kích thước $[227\times227\times3]$.
- CONV1: gồm 96 filters với kích thước là $[11\times11\times3]$ và bước nhảy là 4, pad là 0.
- Các giá trị thuộc filter này được tạo ngẫu nhiên.
- Output: Activation map $\mathbf{M_2}$ có 96 feature map kích thước $[55\times55]$.
- Giải thích:

+ $\mathbf{W_{M_2}} = \frac{(\mathbf{W_{M_1}} - F + 2P)}{S} + 1 = \frac{(227 - 11 + 2 \cdot 0)}{4} + 1 = 55.$
  - $\mathbf{W_{M_1}}$**:** Kích thước của ảnh đầu vào: 227.
  - F : Kích thước của filters: 11.
  - P : Padding: 0.
  - S : Bước nhảy của filters (Stride): 4.
  - $\mathbf{W_{M_2}}$**:** Kích thước của activation map đầu ra: 55.

– ReLU activation function: $f(x) = max(0,\ x)$
– $\mathbf{M_2} = f(\mathbf{M_2}) = max(0,\ \mathbf{M_2})$
– Cụ thể:
  + $\mathbf{M_2}$**:** là activation map có 96 feature map với kích thước [55×55].
  + Hàm ReLU được thực hiện trên tất cả phần tử nằm trong $\mathbf{M_2}$**.**
  + Giá trị trả về lại cho $\mathbf{M_2}$**.**

1.1.4 NORM1

– Local Response Normalization (LRN):

$$b_{x,y}^{i} = \frac{a_{x,y}^{i}}{\left(k + \alpha \sum_{j=max\left(0, i-\frac{n}{2}\right)}^{min\left(N-1, i+\frac{n}{2}\right)} \left(a_{x,y}^{j}\right)^2\right)^{\beta}}$$

– Giải thích:
  + $a_{x,y}^{i}$: Sự hoạt động của 1 neuron tại filter i ở vị trí (x, y).
  + $b_{x,y}^{i}$: Sự hoạt động sau khi chuẩn hóa của 1 neuron tại kernel i ở vị trí (x, y).
  + Những siêu tham số cố định:
    - $k = 2$: Hằng số điều chỉnh độ lớn của mẫu chuẩn hóa. Có chức năng giúp tránh giá trị chuẩn hóa trở nên quá nhỏ giúp ổn định quá trình chuẩn hóa.
    - $n = 5$: Số filters lân cận của filter i.
    - $\alpha = 10^{-4}$: Hằng số điều chỉnh ảnh hưởng của các giá trị hoạt động đến mẫu chuẩn hóa. Giá trị α nhỏ hơn làm cho các hoạt động trước đó có ảnh hưởng lớn hơn đến kết quả chuẩn hóa, giúp kiểm soát mức độ "phạt" cho các giá trị lớn hơn.

- $\beta = 0.75$: Hằng số điều chỉnh độ dốc của hàm chuẩn hóa. Điều chỉnh cách thức mẫu chuẩn hóa ảnh hưởng đến các giá trị hoạt động.

+ Ý nghĩa: Sự hoạt động của 1 neuron tại filter i ở vị trí (x, y) được chia cho sự hoạt động của các neurons tại các filters lân cận ở vị trí (x, y) với các tỷ lệ của hằng số $k,\ n,\ \alpha,\ \beta$.

1.1.5 MAX POOL1

- Input: Tensor $\mathbf{M_2}$ có 96 feature map kích thước [55×55].
- MAX POOL1: Sử dụng filter kích thước [3×3] với bước nhảy là 2 trên tất cả các filters trên activation map $\mathbf{M_2}$.
- Output: Tensor $\mathbf{M_3}$ có 96 feature map kích thước [27×27].
- Giải thích:

+ $\mathbf{W_{M_3}} = \frac{(\mathbf{W_{M_2}} - F + 2P)}{S} + 1 = \frac{(55-3+2\cdot 0)}{2} + 1 = 27.$

  - $\mathbf{W_{M_2}}$**:** Kích thước của tensor đầu vào: 55.
  - F : Kích thước của filters: 3.
  - P : Padding: 0.
  - S : Bước nhảy của filters (Stride): 2.
  - $\mathbf{W_{M_3}}$**:** Kích thước của tensor đầu ra: 27.

1.1.6 CONV2

- Input: Tensor $\mathrm{M_3}$ có 96 feature map kích thước [27×27].
- CONV2: gồm 256 filters với kích thước [5×5×96] và bước nhảy là 1, pad là 2.
- Các giá trị thuộc filter này được tạo ngẫu nhiên.
- Output: Activation map $\mathbf{M_4}$ có 256 feature map kích thước [27×27]
- Giải thích:

+ $\mathbf{W_{M_4}} = \frac{(\mathbf{W_{M_3}} - F + 2P)}{S} + 1 = \frac{(27-5+2\cdot 2)}{1} + 1 = 27.$

  - $\mathbf{W_{M_3}}$**:** Kích thước của tensor đầu vào: 27.
  - F : Kích thước của filters: 5.
  - P : Padding: 2.
  - S : Bước nhảy của filters (Stride): 1.
  - $\mathbf{W_{M_4}}$**:** Kích thước của activation map đầu ra: 27.

- ReLU activation function: $f(x) = max(0,\ x)$
- $\mathbf{M_4} = f(\mathbf{M_4}) = max(0,\ \mathbf{M_4})$
- Cụ thể:

+ **$\mathbf{M_4}$:** là activation map có 256 feature map với kích thước [27×27].
+ Hàm ReLU được thực hiện trên tất cả phần tử nằm trong **$\mathbf{M_4}$.**
+ Giá trị trả về lại cho **$\mathbf{M_4}$.**

1.1.7 NORM2

– Local Response Normalization (LRN):

$$b_{x,y}^{i} = \frac{a_{x,y}^{i}}{\left(k + \alpha \sum_{j=max\left(0,i-\frac{n}{2}\right)}^{min\left(N-1,i+\frac{n}{2}\right)} \left(a_{x,y}^{j}\right)^{2}\right)^{\beta}}$$

– Giải thích:
+ $a_{x,y}^{i}$: Sự hoạt động của 1 neuron tại filter i ở vị trí (x, y).
+ $b_{x,y}^{i}$: Sự hoạt động sau khi chuẩn hóa của 1 neuron tại kernel i ở vị trí (x, y).
+ Những siêu tham số cố định:
  - $k = 2$: Hằng số điều chỉnh độ lớn của mẫu chuẩn hóa. Có chức năng giúp tránh giá trị chuẩn hóa trở nên quá nhỏ giúp ổn định quá trình chuẩn hóa.
  - $n = 5$: Số filters lân cận của filter i.
  - $\alpha = 10^{-4}$: Hằng số điều chỉnh ảnh hưởng của các giá trị hoạt động đến mẫu chuẩn hóa. Giá trị α nhỏ hơn làm cho các hoạt động trước đó có ảnh hưởng lớn hơn đến kết quả chuẩn hóa, giúp kiểm soát mức độ "phạt" cho các giá trị lớn hơn.
  - $\beta = 0.75$: Hằng số điều chỉnh độ dốc của hàm chuẩn hóa. Điều chỉnh cách thức mẫu chuẩn hóa ảnh hưởng đến các giá trị hoạt động.

  Ý nghĩa: Sự hoạt động của 1 neuron tại filter i ở vị trí (x, y) được chia cho sự hoạt động của các neurons tại các filters lân cận ở vị trí (x, y) với các tỷ lệ của hằng số $k,\ n,\ \alpha,\ \beta$.

1.1.8 MAX POOL2

– Input: Tensor **$\mathbf{M_4}$** có 256 feature map kích thước $[27 \times 27]$.
– MAX POOL2: Sử dụng filter kích thước $[3 \times 3]$ với bước nhảy là 2 trên tất cả các filters trên activation map **$\mathbf{M_4}$**.

- Output: Tensor $\mathbf{M_5}$ có 256 feature map kích thước $[13 \times 13]$.
- Giải thích:
  + $\mathbf{W_{M_5}} = \frac{(\mathbf{W_{M_4}} - F + 2P)}{S} + 1 = \frac{(27-3+2\cdot 0)}{2} + 1 = 13.$
    - $\mathbf{W_{M_4}}$**:** Kích thước của tensor đầu vào: 27.
    - F : Kích thước của filters: 3.
    - P : Padding: 0.
    - S : Bước nhảy của filters (Stride): 2.
    - $\mathbf{W_{M_5}}$**:** Kích thước của tensor đầu ra: 13.

1.1.9 CONV3

- Input: Tensor $\mathrm{M_5}$ có 256 feature map kích thước$[13 \times 13]$.
- CONV3: gồm 384 filters với kích thước là $[3 \times 3 \times 256]$.
- và bước nhảy là 1, pad là 1.
- Các giá trị thuộc filter này được tạo ngẫu nhiên.
- Output: Activation map $\mathbf{M_6}$ có 384 feature map kích thước $[13 \times 13]$.
- Giải thích:
  + $\mathbf{W_{M_6}} = \frac{(\mathbf{W_{M_5}} - F + 2P)}{S} + 1 = \frac{(13-3+2\cdot 1)}{1} + 1 = 13.$
    - $\mathbf{W_{M_5}}$**:** Kích thước của tensor đầu vào: 13.
    - F : Kích thước của filters: 3.
    - P : Padding: 1.
    - S : Bước nhảy của filters (Stride): 1.
    - $\mathbf{W_{M_6}}$**:** Kích thước của activation map đầu ra: 13.
- ReLU activation function: $f(x) = max(0,\ x)$
- $\mathbf{M_6} = f(\mathbf{M_6}) = max(0,\ \mathbf{M_6})$
- Cụ thể:
  + $\mathbf{M_6}$**:** là activation map có 384 feature map với kích thước $[13\times13]$.
- Hàm ReLU được thực hiện trên tất cả phần tử nằm trong $\mathbf{M_6}$**.**
  + Giá trị trả về lại cho $\mathbf{M_6}$**.**

1.1.10 CONV4

- Input: Tensor $\mathrm{M_6}$ có 384 feature map kích thước$[13 \times 13]$.
- CONV4: gồm 384 filters với kích thước là $[3 \times 3 \times 384]$.
- và bước nhảy là 1, pad là 1.
- Các giá trị thuộc filter này được tạo ngẫu nhiên.

- Output: Activation map $\mathbf{M_7}$ có 384 feature map kích thước $[13 * 13]$.
- Giải thích:
  + $\mathbf{W_{M_7}} = \frac{(\mathbf{W_{M_6}} - F + 2P)}{S} + 1 = \frac{(13-3+2\cdot1)}{1} + 1 = 13.$
    - $\mathbf{W_{M_6}}$**:** Kích thước của tensor đầu vào: 13.
    - F : Kích thước của filters: 3.
    - P : Padding: 1.
    - S : Bước nhảy của filters (Stride): 1.
    - $\mathbf{W_{M_7}}$**:** Kích thước của activation map đầu ra: 13.
- ReLU activation function: $f(x) = max(0,\ x)$
- $\mathbf{M_7} = f(\mathbf{M_7}) = max(0,\ \mathbf{M_7})$
- Cụ thể:
  + $\mathbf{M_7}$**:** là activation map có 384 feature map với kích thước $[13\times13]$.
- Hàm ReLU được thực hiện trên tất cả phần tử nằm trong $\mathbf{M_7}$**.**

Giá trị trả về lại cho $\mathbf{M_7}$**.**

1.1.11 CONV5

- Input: Tensor M_7 có 384 feature map kích thước$[13 \times 13]$.
- CONV5: gồm 256 filters với kích thước là $[3 \times 3 \times 384]$.
- và bước nhảy là 1, pad là 1.
- Các giá trị thuộc filter này được tạo ngẫu nhiên.
- Output: Activation map $\mathbf{M_8}$ có 256 feature map kích thước $[13 \times 13]$.
- Giải thích:
  + $\mathbf{W_{M_8}} = \frac{(\mathbf{W_{M_7}} - F + 2P)}{S} + 1 = \frac{(13-3+2\cdot1)}{1} + 1 = 13.$
    - $\mathbf{W_{M_7}}$**:** Kích thước của tensor đầu vào: 13.
    - F : Kích thước của filters: 3.
    - P : Padding: 1.
    - S : Bước nhảy của filters (Stride): 1.
    - $\mathbf{W_{M_8}}$**:** Kích thước của activation map đầu ra: 13.
- ReLU activation function: $f(x) = max(0,\ x)$
- $\mathbf{M_8} = f(\mathbf{M_8}) = max(0,\ \mathbf{M_8})$
- Cụ thể:
  + $\mathbf{M_8}$**:** là activation map có 256 feature map với kích thước $[13\times13]$.
  + Hàm ReLU được thực hiện trên tất cả phần tử nằm trong $\mathbf{M_8}$**.**

+ Giá trị trả về lại cho $\mathbf{M_8}$.

1.1.12 MAX POOL3

- Input: Tensor $\mathbf{M_8}$ có 256 feature map kích thước $[13 \times 13]$.
- MAX POOL2: gồm filter kích thước $[3 \times 3]$với bước nhảy là 2 trên tất cả các filters trên activation map $\mathbf{M_8}$.
- Output: Tensor $\mathbf{M_9}$ có 256 feature map kích thước $[6 \times 6]$.
- Giải thích:
  + $\mathbf{W_{M9}} = \frac{(\mathbf{W_{M8}} - F + 2P)}{S} + 1 = \frac{(13-3+2\cdot 0)}{2} + 1 = 6.$
    - $\mathbf{W_{M9}}$**:** Kích thước của tensor đầu vào: 13.
    - F : Kích thước của filters: 3.
    - P : Padding: 0.
    - S : Bước nhảy của filters (Stride): 2.
    - $\mathbf{W_{M8}}$**:** Kích thước của tensor đầu ra: 6.

1.1.13 FC6

- Input: $\mathrm{H}_1 = \mathbf{M_9}.\boldsymbol{flatten}()$
- Output: Hidden layer $\mathbf{H_2}$ kích thước $[4096,]$.
- Giải thích:
  + $\mathbf{H_2} = W * \mathrm{H}_1.$
    - $\mathbf{H_1}$**:** Kích thước của activation map đầu vào sau khi duỗi
    - $W_{4096*9216}$:
    - $\mathbf{H_2}$**:** Kích thước của hidden layer đầu ra: $[4096,]$.
- ReLU activation function: $f(x) = max(0,\ x)$
- $\mathbf{H_2} = f(\mathbf{H_2}) = max(0,\ \mathbf{H_2})$
- Cụ thể:
  + $\mathbf{H_2}$**:** là hidden layer có 4096 neuron.
  + Hàm ReLU được thực hiện trên tất cả phần tử nằm trong $\mathbf{H_2}$**.**
  + Giá trị trả về lại cho $\mathbf{H_2}$**.**
- Dropout:
  + Gán ngẫu nhiên 1 nửa số neuron của $\mathbf{H_2}$ với giá trị 0**.**

1.1.14 FC7

- Input: Hidden layer $\mathbf{H_2}$ kích thước $[4096,]$.
- Output: Hidden layer $\mathbf{H_3}$ kích thước $[4096,]$.
- Giải thích:

+ $\mathbf{H_3} = W * \mathrm{H}_2$.
  - **$\mathbf{H_2}$:** Kích thước của Hidden layer đầu vào
  - $W_{4096*4096}$.
  - **$\mathbf{H_2}$:** Kích thước của hidden layer đầu ra: $[4096,]$.

– ReLU activation function: $f(x) = max(0,\ x)$
– $\mathbf{H_3} = f(\mathbf{H_3}) = max(0,\ \mathbf{H_3})$
– Cụ thể:
  + **$\mathbf{H_3}$:** là hidden layer có 4096 neuron.
  + Hàm ReLU được thực hiện trên tất cả phần tử nằm trong **$\mathbf{H_3}$.**
  + Giá trị trả về lại cho **$\mathbf{H_3}$.**
– Dropout:
  + Gán ngẫu nhiên 1 nửa số neuron của **$\mathbf{H_3}$** với giá trị **0.**

1.1.15 FC8

– Input: Hidden layer **$\mathbf{H_3}$** kích thước $[4096,]$.
– Output: Hidden layer **$\mathbf{H_4}$** kích thước $[1000,]$.
– Giải thích:
  + $\mathbf{H_4} = W * \mathrm{H}_3$.
    - **$\mathbf{H_3}$:** Kích thước của Hidden layer đầu vào
    - $W_{1000*4096}$.
    - **$\mathbf{H_4}$:** Kích thước của hidden layer đầu ra: $[1000,]$.
– Softmax: $z(x) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$
– $\mathbf{H_4} = z(\mathbf{H_4}) = \frac{e^{H_4^i}}{\sum_{j=1}^{1000} e^{H_4^j}}$
– Cụ thể:
  + **$\mathbf{H_4}$:** là hidden layer có 1000 neuron.
  + Hàm Softmax được thực hiện trên tất cả phần tử nằm trong **$\mathbf{H_4}$.**
  + Giá trị trả về lại cho **$\mathbf{H_4}$.**