Attention (cont.)

Arush Iyer

CBAM

- Quick review
  - Channel submodule
    - Apply average and max pooling across each channel
    - Run those through an MLP
    - Apply a sigmoid activation function to get weights between 0 and 1

$$\mathbf{M_c}(\mathbf{F}) = \sigma(MLP(AvgPool(\mathbf{F})) + MLP(MaxPool(\mathbf{F})))$$

  - Spatial submodule
    - Apply average and max pooling
    - Stack average and max maps together
    - Apply a convolutional layer (usually a 7x7 filter size)
    - Apply the sigmoid function

$$\mathbf{M_s}(\mathbf{F}) = \sigma(f^{7\times 7}([AvgPool(\mathbf{F}); MaxPool(\mathbf{F})]))$$

CBAM: End to End Example

- Task: Image classification
- 

CBAM: End to End Example

- Step 0: Feature Extraction
  - Passing an image of size C x H x W into a CNN to get a feature map F
    - C: Channels, H: Height, W: Width
    - $F \in R^{C \times H \times W}$
  - For simplicity, we'll represent the channels as 3x3 matrices

$$F = \begin{bmatrix} 1 & 0 & 1 \\ 2 & 1 & 0 \\ 0 & 2 & 1 \end{bmatrix}_{\text{channel 1 (edges)}}, \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 2 \\ 2 & 0 & 1 \end{bmatrix}_{\text{channel 2 (colors)}}, \begin{bmatrix} 2 & 2 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 2 \end{bmatrix}_{\text{channel 3 (textures)}}$$

CBAM: End to End Example

- Step 1: Channel Submodule
  - Perform average and max pooling for each channel
    - Ex: Channel 1

$$\begin{pmatrix} 1 & 0 & 1 \\ 2 & 1 & 0 \\ 0 & 2 & 1 \end{pmatrix}$$

Matrix Representation (F_1)

Average Pooling

$$\frac{1+0+1+2+1+0+0+2+1}{9} = \frac{8}{9}$$

Max Pooling: $\text{Max}(F_1) = 2$

  - Passed through a two layer MLP

$$\begin{aligned} \mathbf{M_c}(\mathbf{F}) &= \sigma(MLP(AvgPool(\mathbf{F})) + MLP(MaxPool(\mathbf{F}))) \\ &= \sigma(\mathbf{W_1}(\mathbf{W_0}(\mathbf{F^c_{avg}})) + \mathbf{W_1}(\mathbf{W_0}(\mathbf{F^c_{max}}))), \end{aligned}$$

CBAM: End to End Example

- Step 1: Channel Submodule (cont.)
  - Assume learned weights $W_0 = 0.5$, $W_1 = 1.0$
    - Use reduction ratio r so MLP is trained on C/r channels
  - For $F^c_{avg} \approx 0.89$, and $F^c_{max} = 2$, we have
    - $M_c(F)_1 = \sigma(W_1 * \mathrm{ReLU}(W_0 * F^c_{avg}) + W_1 * \mathrm{ReLU}(W_0 * F^c_{max}))$
    - $M_c(F)_1 = \sigma(1 * \mathrm{ReLU}(0.5 * 0.89) + 1 * \mathrm{ReLU}(0.5 * 2))$
    - $M_c(F)_1 = \sigma(0.445 + 1)$
    - $M_c(F)_1 \approx 0.81$
  - Repeat for other channels $M_c(F)_2$ and $M_c(F)_3$
  - Multiply channel wise attention weights back to original feature map F to get $F' \in R^{C \times H \times W}$

ReLU (x) = x if x > 0 else 0

$$F' = \begin{bmatrix} M_c(F)_1 \cdot F_1 \\ M_c(F)_2 \cdot F_2 \\ M_c(F)_3 \cdot F_3 \end{bmatrix}$$

CBAM: End to End Example

- Step 2: Spatial Submodule
  - Perform global pooling across channels for each pixel
    - Average pooling:
      - General: $$F'_{avg}[i,j] = \frac{1}{C}\sum_{c=1}^{C}\left(F'_{c,i,j}\right)$$
      - E.g., pixel [1,1]: $$F'_{avg}[1,1] = \frac{F'_{1,1,1} + F'_{2,1,1} + F'_{3,1,1}}{3}$$
    - Max pooling: $$F'_{max}[i,j] = max_c\left(F'_{c,i,j}\right)$$

CBAM: End to End Example

- Step 2: Spatial Submodule (cont.)
  - Concatenate along channel axis
  - Apply convolutional layer (represented below with kernel size 7x7)

$$\mathbf{M_s}(\mathbf{F}) = \sigma(f^{7\times 7}([AvgPool(\mathbf{F}); MaxPool(\mathbf{F})])) \\ = \sigma(f^{7\times 7}([\mathbf{F^s_{avg}}; \mathbf{F^s_{max}}])),$$

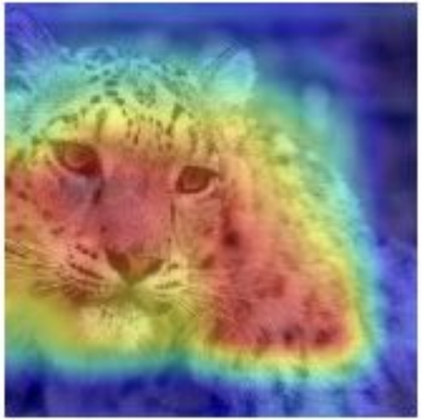
- Step 3: Combination
  - Multiply spatial attention map to every pixel location across all channels
  - $F''_{c,i,j} = M_s(F')_{i,j} * F'_{c,i,j}$
  - $F'' \in R^{C \times H \times W}$

$$\mathbf{F}' = \mathbf{M_c}(\mathbf{F}) \otimes \mathbf{F}, \\ \mathbf{F}'' = \mathbf{M_s}(\mathbf{F}') \otimes \mathbf{F}',$$

CBAM: Results

Input Image

ResNET
P = 0.86



ResNET with CBAM
P = 0.98

Source

Woo, Sanghyun, et al. "CBAM: Convolutional Block Attention Module." Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 3–19. https://arxiv.org/abs/1807.06521