Detailed about FLOPs and Parameter Calculations

Paper Title – "*AC-Lite: A Lightweight Image Captioning Model for Low-Resource Assamese Language.*"

NOTE:

A. MAC: the number of multiplication and addition performed by the CPU/GPU.
B. FLOPs: Floating points operation per sec.
C. A captioning model generates caption words one at a time. If the caption length is set to T, the decoder of each captioning model runs T times to generate words C={w_1, … w_i …., w_T }.
D.
    1. 1 FLOPs = 2 X MAC
    2. 1 GFLOPs = FLOPS x 10^9
    3. 1 MFLOPs = FLOPS x 10^6

LSTM FLOPs :

FLOPs : 4 (input_size X hidden_size + hidden_size)

Here,

1. input_size = [prev_h, fc_feats, xt] if Attention LSTM
2. input_size = [att, h_att] if Language LSTM

Parameters : **4 (input_size X hidden_size + hidden_size^2 + hidden_size)**

GRU FLOPs :

Formula : 3 (input_size X hidden_size + hidden_size)

Here,

1. input_size = [prev_h, fc_feats, xt] if Attention GRU
2. input_size = [att, h_att] if Language GRU

Parameters : **3 (input_size X hidden_size + hidden_size^2 + hidden_size)**

AOANet FLOPs (MULTI HEAD) :

1. **Total FLOPs = $H * (10\ Nd^2 + 2N^2d) + 2ND^2 + D$**
2. **Total parameters=6*(D*D)+6*D**

FLOPs for a Single Encoder Stack of Transformer:

NOTE:

N → sequence length

D → Hidden size

D = D/H (attention head dimension)

1. **Total FLOP for Single Encoder Block**

   $H * (10\ Nd^2 + 2N^2d) + 16ND^2 + 2ND$
2. **Total parameters= $4D^2 + 2D \times D_{ff}$**

   Here, D_{ff} is the dimension of FFN Layer.

FLOPs for a Single Decoder Stack of Transformer :

NOTE :

N → Encoder sequence length

L → Decoder sequence length

D- Hidden size

d=D/H (attention head dimension)

3. **Total FLOP for Single Decoder Block**
   a. **$= H * (12\ Ld^2 + 2L^2d + 2Nd^2 + 4LNd) + 16ND^2 + 3ND$**
4. **Total parameters= $8D^2 + 2D \times D_{ff}$**

   Here, D_{ff} is the dimension of FFN Layer.

FLOPs of Faster R-CNN:

Visual Encoder: Faster R-CNN object detector working Encoder in image captioning. To compute the GFLOPS for faster r-cnn we use the standard size (VGA image resolution) of object detection. The Faster R-CNN typically consists of:

1. **Feature Extractor Backbone of Faster R-CNN**
2. **RPN (Region Proposal Network)**
3. **Classification and Regression Layer of Proposal**

1. Feature Extractor Backbone: In the image captioning encoder, ResNet101 is typically employed within Faster R-CNN, utilizing only the convolutional layers of ResNet101. As a result, the feature extractor involves a computational cost of **96.1** GFLOPs and 42.5 Million Parameters.

2. **RPN (Region Proposal Network)**: RPN uses three convolution layers, which are:

```
conv): Conv2d(1024, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
(objectness_logits): Conv2d(512, 12, kernel_size=(1, 1), stride=(1, 1))
(anchor_deltas): Conv2d(512, 48, kernel_size=(1, 1), stride=(1, 1))
```

So, GFLOPs is 11.398 GFLOPs and 4.749 million

3. **Classification and Regression Layer of Proposal:** In Faster R-CNN, 300 proposals from RPN were used for classification and regression. Each classification and regression 0.032 GFLOPs, which are:

```
(box_predictor): FastRCNNOutputLayers(
        (cls_score): Linear(in_features=2048, out_features=1601, bias=True)
        (bbox_pred): Linear(in_features=2048, out_features=6400, bias=True)
)
```

So, 300 proposal GFLOPs is 9.831 GFLOPs, and the parameter 16.386 million (considers only one proposal)