Show Attend Tell - PyTorch Implementation

1 Data Pre-Processing

1. First we tokenised all the captions using *RegexpTokenizer* from NLTK library.
2. We built the dictionary of the words that we have using only the train dataset.
3. Assigned an unique integer id to every word in our dictionary.
4. Assigned an ID - 0 to $< start >$ token, 1 to $< eos >$(end of sentence) token, 2 to $< unk >$(unkown) token, 3 to $< pad >$(padding) token.
5. Then converted the tokeinsed words in the caption to their id and added $< start >$ token to the start of every caption and $< eos >$ token to the end of every caption.
6. Added padding to the captions by adding the $< pad >$ token to every caption whose length is less than the length of the max length of caption in the train dataset.
7. For the images in the dataset, we size all the images to (224,224) and normalise it to using $mean = [0.485, 0.456, 0.406], std = [0.229, 0.224, 0.225]$ and use a batch of size 64 to train the model.

2 Methodology

1. For training the caption generation model, we first extract the images features of shape (14,14,512) from an image of size (224,224,3) using pre-trained VGG-19.
2. We we keep the starting conv layers of VGG-19 layers as fixed and fine tune the rest of the layers over the course of training our model, because the first layer are help full in extracting vital features of the image, which we dont want to lose by fine tuning.
3. Then we use the extracted image features to get the initial hidden state and cell state of the LSTM cell of shape (512) for each image, by using a linear layer with tanh activation function.
4. Then we generate the caption word by word by :
   (a) Calculating the soft attention using the image features and previous output thought a linear layer, followed by tanh activation.
   (b) Then using a sigmoid activated linear transform of the LSTN's previous hidden state and the attention, embedding (size-512) of the previous generated words, we generate a new output embedding of size 512.
   (c) Then we pass the output of the LSTM cell through a deep output layer of $input_dim = 512$ and $output_dim$ =size of the vocabulary to get the new generated word.