Deep Learning(CS60010)

Term Project
Group-46
TRANSFORMERS

Abothula Suneetha (20CS10004)
Shreya Agrawal (20CS10058)
Priyanshi Dixit (20CS30069)

Part - A

We present the architecture of an image captioning model consisting of an EncoderCNN and a DecoderRNN.

EncoderCNN: The EncoderCNN utilizes a pre-trained ResNet-50 model to extract features from images. The ResNet-50 model is truncated, excluding the final fully connected layer. The extracted features are then passed through a linear layer followed by batch normalization to obtain a fixed-size feature vector. This feature vector serves as the input to the decoder for generating captions.

DecoderRNN: The DecoderRNN is responsible for generating captions based on the features extracted by the EncoderCNN. It consists of an LSTM (Long Short-Term Memory) network with a specified number of layers. The LSTM takes the image features along with the caption embeddings as input and produces a sequence of hidden states. These hidden states are then linearly transformed to obtain the output scores for each word in the vocabulary. The word with the highest score is selected as the predicted word at each time step. This process continues until the end token is generated or a maximum length is reached.

An EncoderCNN is defined with an embedding size of 512.A DecoderRNN is defined with an embedding size of 512, hidden size of 512, and a vocabulary size (3297) determined by the length of the vocabulary calculated using training captions. Cross-entropy loss is used as the loss function. Adam optimizer is used with a learning rate of 0.001.

For each batch in the training data loader:The optimizer gradients are zeroed. Image features are extracted using the encoder. Captions are passed through the decoder, and outputs are generated. The loss is computed between the predicted outputs and the ground truth captions. Backpropagation is performed, and optimizer steps are taken. Training statistics such as loss and accuracy are updated.

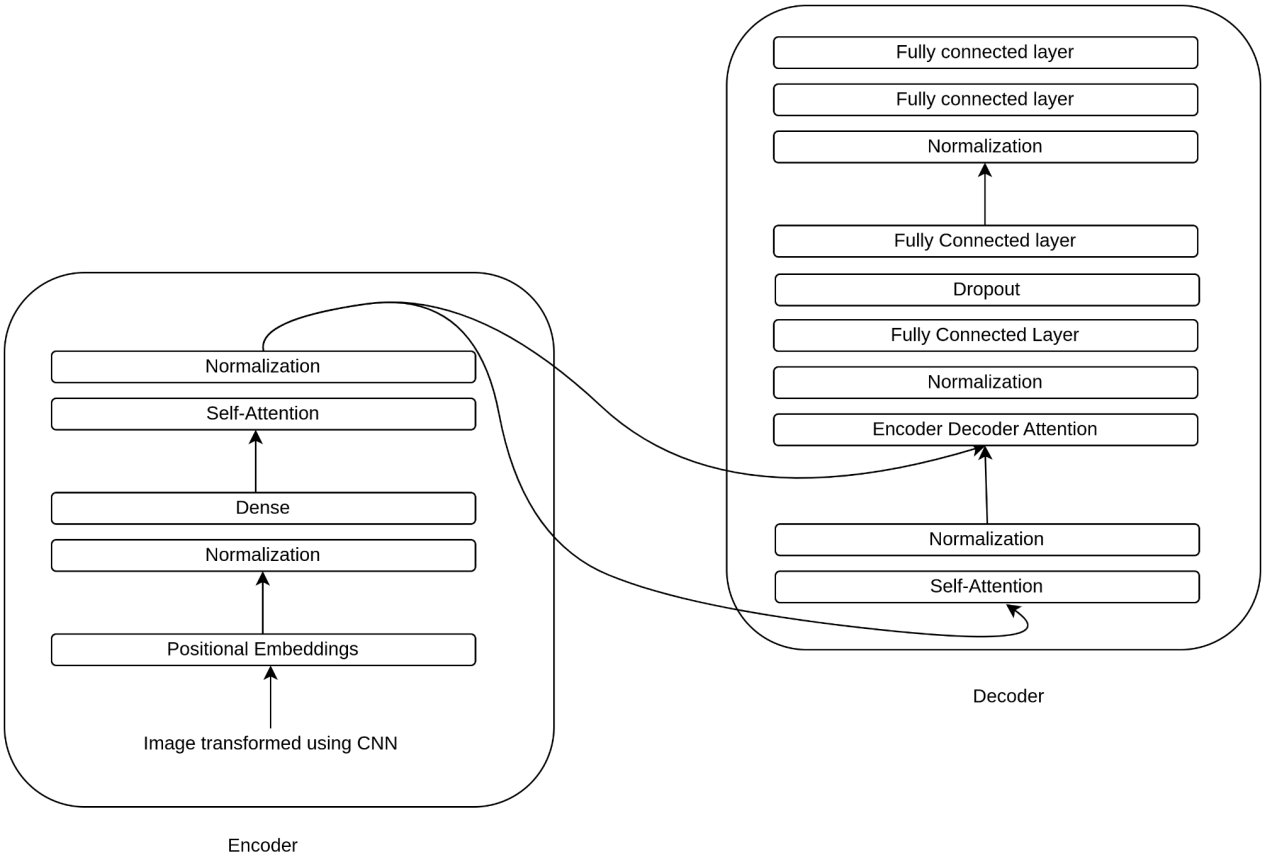
After each epoch, the model is evaluated on the validation set: The encoder and decoder are set to evaluation mode. Validation loss is computed similarly to the training loss. The best model is saved based on the validation loss.

Due to time and resource constraints, we were not able to run for more number of epochs, we ran it for 1 epoch and ***Train Loss: 3.1296, Accuracy: 0.5641. Validation Loss: 2.3183***

If the model is trained for a number of epochs, say 20, the train loss gradually decreases and the model learns better.

Part - B

Methodology



Encoder: Images are transformed using CNN, a pretrained model imagenet, embedded with positional embeddings, and then sent to a normalization layer, followed by an FC layer. The encodings go through the self-attention layer and normalization. The output is then used as an input for the decoder and for encoder-decoder attention

Decoder: The decoder input is self-attended and normalized, then passes through the encoder-decoder attention layer followed by multiple fully connected and dropout layers for caption generation

Results

- Training accuracy: 0.4115
- Training loss: 2.7254
- Validation accuracy: 0.3291
- Validation loss: 3.5956

Hyperparameters

- **IMAGE SIZE = (299, 299)**
- **VOCABULARY SIZE = 10000**
- **SEQUENCE LENGTH = 500**
- **EMBEDDING DIMENSION = 512**
- **HIDDEN LAYER DIMENSION = 512**
- **BATCH SIZE = 6**
- **EPOCHS=30**