CPSC 8430 DEEP LEARNING HOMEWORK 2

SREERAM PALADUGU

Github-https://github.com/sreerampaladugu10/deep-learning-HW2

Introduction:

The S2VT (Sequence to Sequence with Video to Text) model is used for generating captions for videos.

The data pre-processor script prepares the input data for the model by extracting features from the video frames and generating the corresponding captions. These features are then used as input to the model.

The S2VT model architecture, which contains an encoder layer to process the input features and a decoder layer to provide output captions, is defined in the basic model script. The model also has an attention layer that aids the decoder in concentrating on particular elements of the input when producing captions.

The training script trains the S2VT model on the input data using backpropagation and gradient descent. The model is trained to minimize the difference between generated and actual captions.

The testing script takes a trained S2VT model and generates captions for test video inputs. The output captions are then evaluated using the BLEU score, which compares the generated captions with the actual captions to determine their similarity.

To make it easy to run the testing script, a shell script called Hw2_seq2seq.sh is provided.

Data Pre-processing-

All of the training video features and their accompanying training video captions are read by the data pre-processor script and stored in a feature dictionary and a caption dictionary, respectively. All of the English-language captions for the videos are included in the dictionary. The word will be entered into the dictionary if its usage surpasses the cutoff point (more than three repetitions). To keep the dictionary size small and only record the essential information, words that appear fewer times than the threshold will be stored in the dictionary as <UNK>. We perform this procedure because we cannot feed the input data to the model directly. So that the model can use the input data, These phrases are changed into words and assigned number indices.

data pre-processor script uses four main tokens-

- <PAD>: used to pad the sentences to the same length to maintain uniformity.
- <BOS>: Beginning of the sentence, an identifier is to generate the output sentence.
- <EOS>: End of sentence, an identifier to signal the end of the output sentence.
- <UNK>: This token is used when the word is not stored in the dictionary based on its appearance, so these words will be treated as unknown.

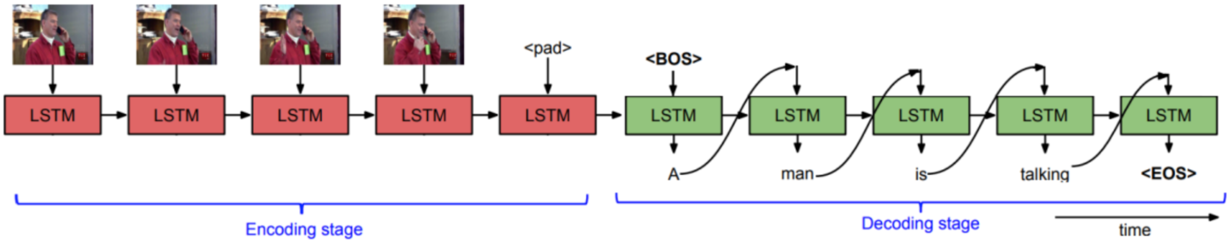
Based on the video ids, the Dataprocessor function loads the video labels and their corresponding features. After feeding the captions into a dictionary, it creates a tensor output for the video features and their captions.

Model Architecture-

The model architecture comprises several parts, which are -

- The BaseModel(S2VT)
- Encoder Layer
- Decoder layer
- Attention Layer
- BLEU score

The encoder and decoder layer script implements the model's encoding and decoding layer. The sequence-to-sequence encoder-decoder model is implemented in the script using two layers of GRU (RNNs). To enhance the efficiency of the underlying model, the Attention Layer script adds an attention layer to the encoder's hidden states. The model can examine a variety of input locations at every decoding time step. The output of the encoder and the hidden state of the decoder are merged as a matching function to create a scalar, which is then softmax processed before the final new hidden state of the decoder is passed to the following stage.



Attention layer-

The attention mechanism in the S2VT model improves alignment between input video frames and generated captions. At each decoding step, the attention layer computes weights for each input feature based on their relevance to the decoder state. These weights are used to compute a context vector, which is then used as input to the decoder at the next step.

The attention mechanism can improve model performance by selectively focusing on different input features, particularly for longer input sequences. It is a key component of state-of-the-art models for sequence-to-sequence tasks.

Training and Testing-

The training is done for the model using the following parameters:
BatchSize= 10
Epochs= 100
Learningrate= 0.0001
LossFunction= nn.CrossEntropyLoss()
Optimizer = Adam
TrainingSampleSize = 1450
Videofeaturesdimension = 4096
Videoframedimension = 80

Using the model produced by the testing. For the test videos, the test function will produce the anticipated caption. The created output caption is then contrasted with the ground truth caption, which produces a BLEU score.

The aforementioned model produced a BLEU score of 0.709