Deep Learning - CPSC 8430
HomeWork -2 Report
By
Chaitanya Mandru

Introduction:

In this homework, a sequential-to-sequential model is presented, trained, and tested in order to generate a caption for videos. The basic idea of Video Caption Generation is to input a video and get a stream of captions for the actions happening in the video.

Generating video captions automatically with natural language has been a challenge for both the field of natural language processing and computer vision [Sequence to Sequence Model for Video Captioning]. In visual interpretation, Recurrent Neural Networks (RNNs), which model grouping elements, have been ended up being proficient. Picture depiction will attempt to deal with a variable-length yield grouping of words, while a video portrayal needs to deal with a variable-length input arrangement of edges as well as a variable-length yield succession of words.

LSTMs have been ended up being effective in seq-to-seq undertakings like Speech acknowledgment and Machine interpretation. Encoding is finished by a stacked LSTM which encodes the casings individually. The result of a Convolutional Neural Network applied to each information casing's power values is utilized as a contribution to the LSTM. The model produces a sentence word by word when every one of the casings is perused.

This homework is achieved by following deep learning techniques and training the dataset by following the requirements:

- Python 3.6.0
- cuda==9.0
- TensorFlow-gpu==1.15.0
- numpy==1.14

More about Dataset:

The dataset we utilized in this examination is MSVD (Microsoft Video Description) dataset. The Microsoft Research Video Description Corpus (MSVD) dataset comprises around 120K sentences. The Dataset contains 1550 video snippets each ranging from 10 to 25 seconds. We have taken the split to 1450 and 100 videos for training and testing respectively.
We have stored the features of each video in the format of 80×4096 after preprocessing using pre-trained CNN VGG19. During, the training we don't reduce the number of frames from the video to take full advantage.

Model Approach:

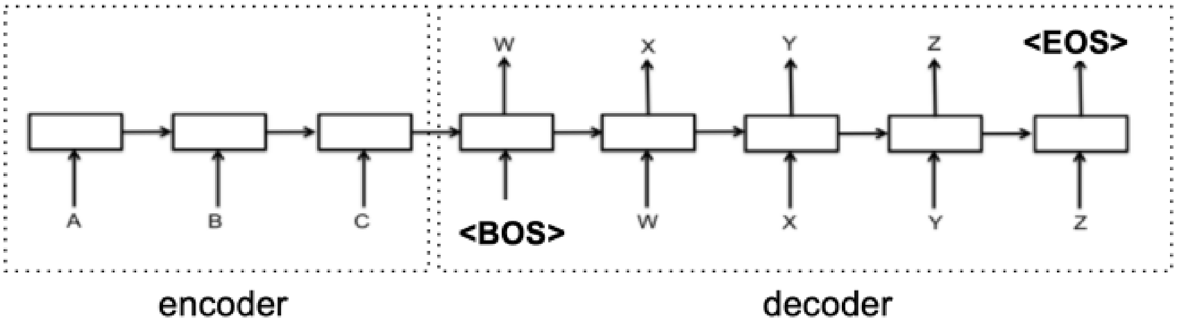
To implement the seq2seq model, first, a dictionary function is created to convert words to indices and vice versa.

- **Dictionary** - most frequently a word or min count

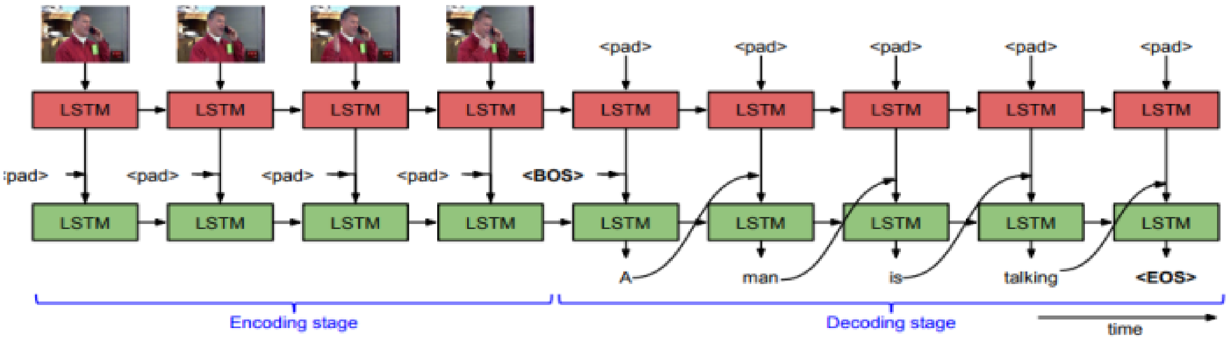
- **Tokenization:**
  <Pad>: pad the sentence to the same length
  <BOS>: Begin of the sentence, a sign to generate the output sentence.
  <EOS>: End of sentence, a sign of the end of the output sentence.
  <UNK>: Use this token when the word isn't in the dictionary or just ignore the unknown word.

The standard for the seq2seq model (S2VT) was carried out. In this model 2 layers of GRU (RNNs) are executed, in the main layer of the RNN, the Video is handled and encoded (encoderRNN), and a result is created with the assistance of the decoder (decoder RNN).

In the interpreting system, tokens are utilized to fragment the subtitles in light of the start and finishing stanzas and perform handling over the video to create the real words.



The below figure illustrates the processing of encoding the video using encoderRNN (a GRU layer) and generating video using decoder RNN (a GRU layer).



Results:

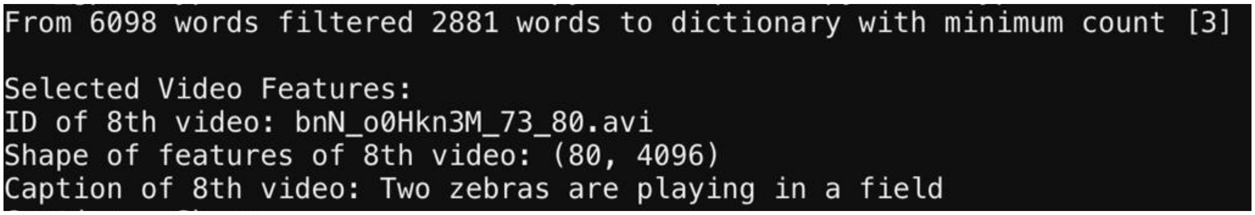
The following commands are used for the execution of the training dataset

Python sequence.py /home/cmandru/chai//MLDS_hw2_data/training_data/feat/ /home/cmandru/chai/MLDS_hw2_data/training_label.json

Parameters that are used while training the model are as follows:

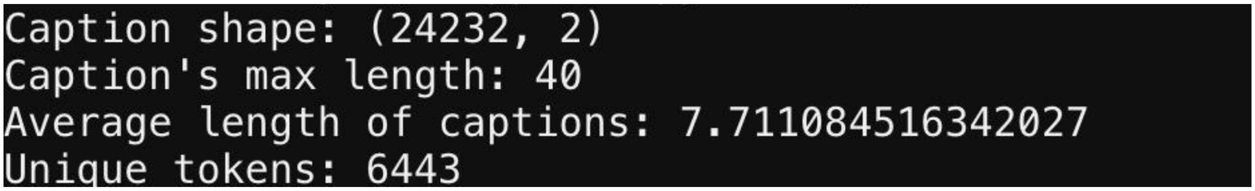
- Learning Rate was set to 0.001
- The batch size was set to 50
- The maximum Gradient norm is 5.0
- No of the hidden units per layer is 1024

Below attached screenshots are the results that I got after successful execution.



```
From 6098 words filtered 2881 words to dictionary with minimum count [3]

Selected Video Features:
ID of 8th video: bnN_o0Hkn3M_73_80.avi
Shape of features of 8th video: (80, 4096)
Caption of 8th video: Two zebras are playing in a field
```

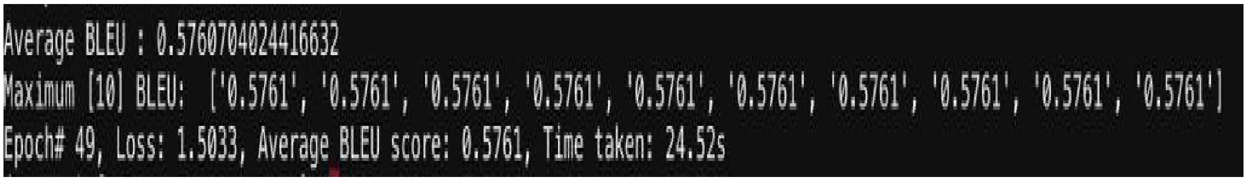


```
Caption shape: (24232, 2)
Caption's max length: 40
Average length of captions: 7.711084516342027
Unique tokens: 6443
```

After training the dataset I used the following commands to test the dataset. I used sequence.py to build the sequence-to-sequence model and train.py to train the model.

python train.py /home/cmandru/chai/MLDS_hw2_data/testing_data/feat/ /home/cmandru/chai/MLDS_hw2_data/testing_label.json ./chai_output.txt

Below is the attached screenshot of the results that I got after training the model.



```
Average BLEU : 0.5760704024416632
Maximum [10] BLEU:  ['0.5761', '0.5761', '0.5761', '0.5761', '0.5761', '0.5761', '0.5761', '0.5761', '0.5761', '0.5761']
Epoch# 49, Loss: 1.5033, Average BLEU score: 0.5761, Time taken: 24.52s
```

BLEU Score:

Bilingual Evaluation Understudy is a calculation for assessing the nature of the text which has been machine-deciphered starting with one regular language then onto the next. Quality is viewed as the correspondence between a machine's result and that of a human: "the nearer a machine interpretation is to an expert human interpretation, the better it is" - this is the focal thought behind BLEU.

BLEU was one of the principal measurements to guarantee a high connection with human decisions of quality and stays perhaps the most famous mechanized and economical measurement. I got the Average BLEU score of around 0.67321 after running the model.