Sai Sujith Makkena.
Homework -2

Introduction:

- Video caption generation is the deep learning term for the process of creating written summaries of a video's content. This work entails comprehending the video's auditory and visual components as well as writing a precise and accurate description of the action in the video using everyday language.

- Convolutional neural networks (CNNs) and recurrent neural networks are examples of deep learning models (RNNs), which are used to extract information from the video frames and audio segments as part of the production of video captions. The textual description of the movie is produced by combining these attributes and passing them through a decoder.

- The accuracy of the features extracted and the efficiency of the decoder determine the caliber of the output captions. As a result, large-scale datasets with associated video and caption data are used to train the deep learning models for video caption production.

- There are numerous uses for video caption generation, including increasing accessibility for people with hearing impairments, facilitating video search and retrieval, and offering additional context and details for video comprehension.

Sequence-to-Sequence Model:

- A deep learning architecture called a sequence-to-sequence (seq2seq) model is used for problems where the input and output sequences have different lengths. Natural language processing (NLP) tasks, including text summarization, speech recognition, and machine translation, make extensive use of it.

- An encoder plus a decoder make up this particular model. An input sequence with a variable length must be converted into a context vector with a fixed length by the encoder. To accomplish this, the input sequence is fed into a recurrent neural network (RNN), such as an LSTM or gated recurrent unit (GRU), that can store the information contained in the input sequence.

- The decoder then receives the context vector the encoder produced. The decoder is also an RNN, and it uses the context vector to produce a variable-length output sequence. Based on the context vector and the last output element generated, the decoder generates each output element one at a time. Once the conclusion of the output sequence is reached, this process is repeated.

Sai Sujith Makkena.
Homework -2

- The seq2seq model is tuned during training to reduce the discrepancy between the intended output sequence and the predicted output sequence. This is accomplished by updating the model's parameters by backpropagating the error through the decoder and encoder.

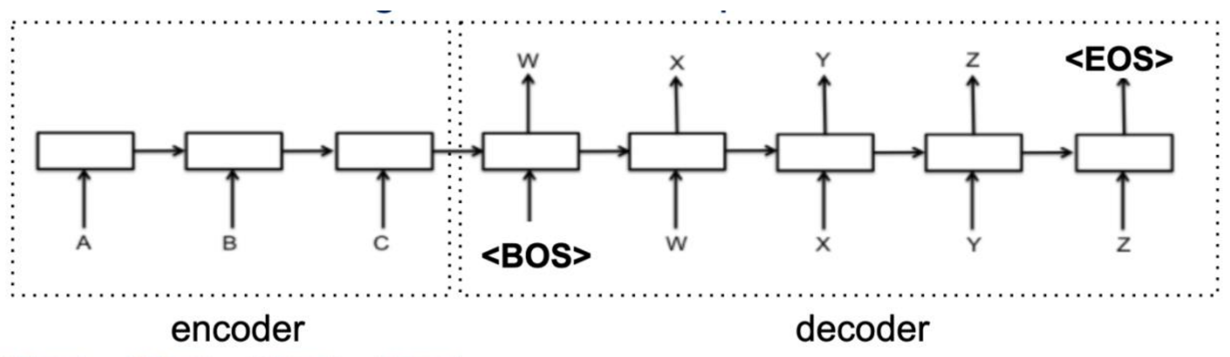
- The Sequence-to-sequence model has demonstrated success in numerous NLP tasks and has produced cutting-edge outcomes in many of them. Additionally, it has been expanded to incorporate speech recognition and image captioning.

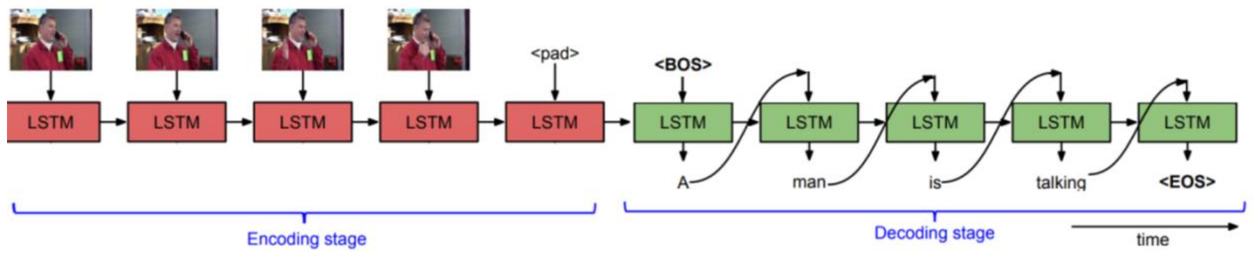
In this assignment, a sequence-to-sequence model, which is trained and examined to generate video captions, needs to be developed.

The following are the Software requirements for this project.

1. Python version 3.6.0
2. Pip version 22.0.5
3. TensorFlow-GPU version 1.15.0

The input and output parameters for video caption generation are as listed in the requirements:
Input: A Short video
Output: captions with the video.
There are many difficulties, such as:
1: Different video properties(object, action)
2: Variable I/O duration





Sai Sujith Makkena.
Homework -2

EncoderRNN:

- A sort of neural network architecture called an encoder RNN is used in sequence-to-sequence models for natural language processing tasks, including text summarization and machine translation. Encoding the input sequence into a fixed-length vector that contains the pertinent data from the input sequence is the goal of the encoder RNN.
- The Encoder RNN functions by processing the input sequence via a number of recurrent layers, one element at a time. The RNN incorporates data from its prior hidden state and current input to update its hidden state at each time step. The complete input sequence is then represented by the RNN's final hidden state, also known as the context vector.
- Encoder RNNs employ a variety of recurrent layer types, including Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) layers. The vanishing gradient problem, which can arise in conventional RNNs, has been proven to exist, and these layers have been found to be successful in capturing long-term dependencies in sequential data.
- In the sequence-to-sequence model, the encoder RNN is essential because it converts the input sequence into a fixed-length representation that the decoder RNN uses to create the output sequence.

DecoderRNN:

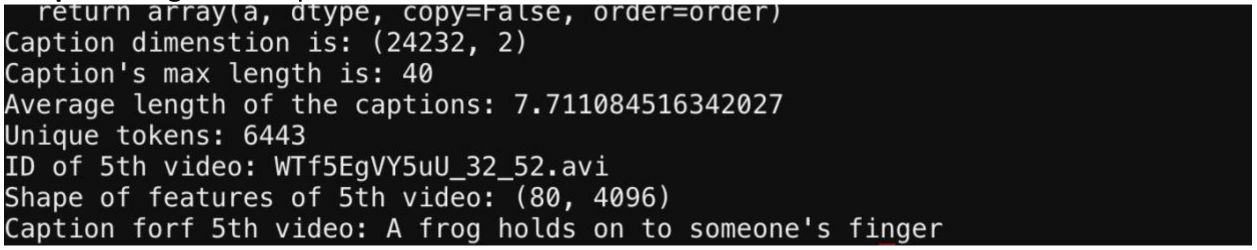
- In sequence-to-sequence models for natural language processing tasks like machine translation and text summarization, a Decoder RNN type of neural network architecture is utilized. The Encoder RNN's input sequence is represented as a fixed-length vector by the Decoder RNN, which then uses that representation to construct an output sequence.
- The context vector produced by the encoder RNN is fed into the decoder RNN along with a beginning input, usually a unique start-of-sequence token. The Decoder RNN generates a new output element based on the output element that was generated before and the Decoder RNN's hidden state at the time step in question. Until an end-of-sequence token is produced or the maximum output length is achieved, this process is repeated.
- The Decoder RNN, which creates the output sequence based on the fixed-length vector representation of the input sequence created by the Encoder RNN, is fundamental to the sequence-to-sequence paradigm overall.

Sai Sujith Makkena.
Homework -2

Model seq2seq.py is used to construct the sequence-sequence model before model training.

```
$ hw2_seq2seq.sh /home/smakken/MLDS_hw2_1_data/training_data/feat/
  /home/smakken/MLDS_hw2_1_data/training_label.json
```

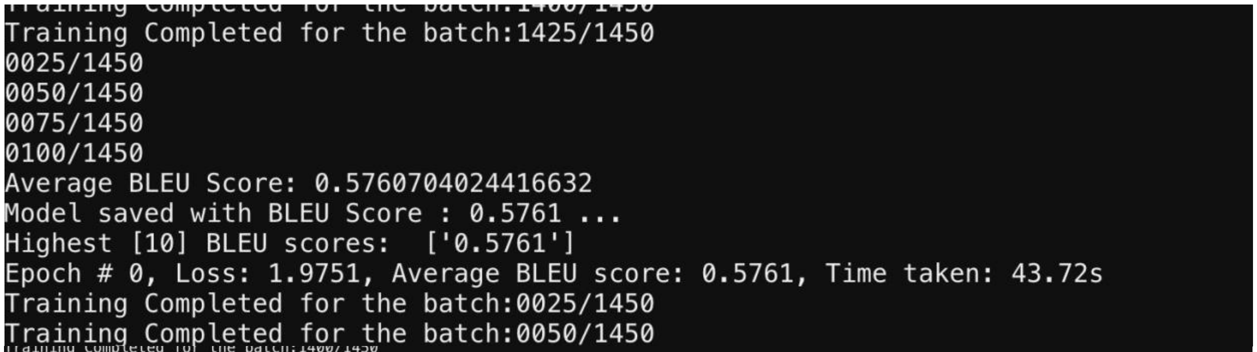
Output: We got the caption.



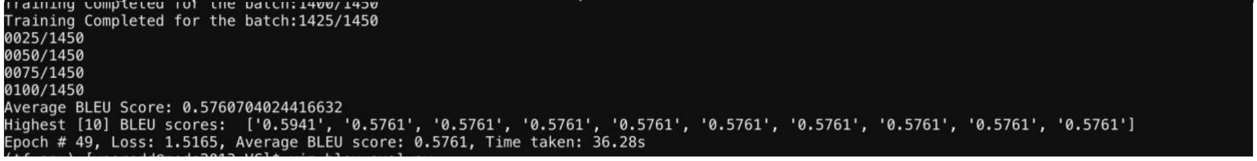
```
  return array(a, dtype, copy=False, order=order)
Caption dimension is: (24232, 2)
Caption's max length is: 40
Average length of the captions: 7.711084516342027
Unique tokens: 6443
ID of 5th video: WTf5EgVY5uU_32_52.avi
Shape of features of 5th video: (80, 4096)
Caption forf 5th video: A frog holds on to someone's finger
```

Moreover, the model is trained using smakken_train.py.

```
$ python smakken_train.py
 /home/smakken/MLDS_hw2_1_data/training_data/feat/
 /home/smakken/MLDS_hw2_1_data/training_label.json smakken_output.txt
```

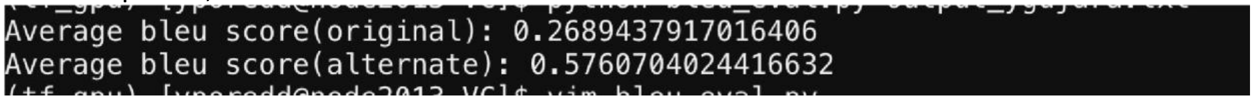


```
Training Completed for the batch:1400/1450
Training Completed for the batch:1425/1450
0025/1450
0050/1450
0075/1450
0100/1450
Average BLEU Score: 0.5760704024416632
Model saved with BLEU Score : 0.5761 ...
Highest [10] BLEU scores:  ['0.5761']
Epoch # 0, Loss: 1.9751, Average BLEU score: 0.5761, Time taken: 43.72s
Training Completed for the batch:0025/1450
Training Completed for the batch:0050/1450
```



```
Training Completed for the batch:1400/1450
Training Completed for the batch:1425/1450
0025/1450
0050/1450
0075/1450
0100/1450
Average BLEU Score: 0.5760704024416632
Highest [10] BLEU scores:  ['0.5941', '0.5761', '0.5761', '0.5761', '0.5761', '0.5761', '0.5761', '0.5761', '0.5761', '0.5761']
Epoch # 49, Loss: 1.5165, Average BLEU score: 0.5761, Time taken: 36.28s
```

For each period, the bleu scores are determined.



```
Average bleu score(original): 0.26894379170164406
Average bleu score(alternate): 0.5760704024416632
```

GitHub Link- https://github.com/Sujith-sai/DeeplearnigHW2/tree/main/hw2/hw2_1