CPSC 8430: Deep Learning Assignment 2
Video Caption Generation – PRAMOD KARKHANI

GitHub link
https://github.com/pramod-karkhani/CPSC-8430-Deep-Learning---Hw2

Problem Statement
Generate a video caption for an input video using sequential -to- sequential model. The input for the code will be a video. The output will be a stream of captions describing in short the actions/events occuring in the video.

The above is achieved by Recurrent Neural Networks.
Requirements
• Python
• CUDA
• torch
• numpy
• scipy
• pickle
• pandas

Dataset
The dataset provided as part of the Homework requirement has been utilised to train and test the model.
The dataset contains 1450 videos for training & 100 for the purpose of testing. Training and testing label.json files are also present which provided us with labels to train the model.

Approach:
Our Model is a seq-to-seq model which takes sequence of video frames as input
and generate sequence of words as output

Padded sequences and maskings are applied to the data during the preprocessing stage.
Using packed padded sequences, we may limit the processing of our RNN to the input sentence's non-padded components.
By using a method called masking, we can prevent the model from detecting things like padding components that we don't want it to.This procedure is used to raise performance.
The MLDS hw2 1 data folder on the local PC contains the training and testing data that were taken from the HW PPT. Minimum vocabulary size should be 3.

Tokenization:
1. <pad>: Pad the sentence to the same length
2. <bos>: Begin of sentence, a sign to generate the output sentence.
3. <eos>: End of sentence, a sign of the end of the output sentence.
4. <unk>: Use this token when the word is not found in the dictionary /just ignore the unknown word.

The dictionary of id and caption of respective videos were stored in 2 object files. The dictionary is being built using the object files '*vid id.obj*',' *dict caption.obj*', and' *dict feat.obj*'.

sequence.py is used for execution, at node (Palmetto cluster), using the following command

python sequence.py
/home/pkarkha/CPSC_8430/MLDS_hw2_1_data/training_data/feat/
/home/pkarkha/CPSC_8430/MLDS_hw2_1_data/training_label.json
I got the following results:

From 6098 words filtered 2881 words to dictionary with minimum count -3
Caption dimension: (2432,2)
Caption's max length: 40
Average length of captions: 7.77108
Unique tokens: 6443

The next task is to train the model we created. There are two .py files here. We'll use sequence.py to build the required seq-to-seq model. Then, With the help of train.py, we will train the model. I used the hyperparameters mentioned in the HW PPT.
I ran the following command in palmetto cluster on my active node using ssh.
python train.py /home/pkarkha/CPSC_8430/MLDS_hw2_1_data/training_data/feat/
/home/pkarkha/CPSC_8430/MLDS_hw2_1_data/training_label.json
./output_testset.txt

There are 3 Args in the above command. The first Arg is the path of the feature map which are in the format of .npy file and second Arg is the path of the testing_label.json file which has captions for a particular video id

Results:
I Finally calculated the Bleu scores
An Average bleu score of 0.6766 was reached after 6 epochs.
After training the model for 200 epochs, the score reached ~ 0.662