Analyzing Hyperparameters for a CNN LSTM Video Description Framework

Ruth Okoilu
Computer Science
NC State University
Raleigh, NC
rookoilu@ncsu.edu

Rohit Naik
Computer Science
NC State University
Raleigh, NC
rtnaik@ncsu.edu

Nirav Jain
Computer Science
NC State University
Raleigh, NC
najain@ncsu.edu

Udit Deshmukh
Computer Science
NC State University
Raleigh, NC
udeshmu@ncsu.edu

***Abstract*—In this paper, we explore a deep learning encoder-decoder framework that describes objects or/and activities in a video by modelling the local and global temporal structure of videos. The framework consists of deep Convolutional Neural Networks (CNNs) which serve as an encoder and Long Short-Term Memory Network (LSTM) for the decoder. This design is effective for producing high learning capacity models for video to text translations as long as critical hyper-parameters are identified and provided with appropriate values. Our study therefore seeks to discover the pivotal hyper-parameters and experimental conditions in which this approach works best. The framework for our experiments provides description for the Youtube2Text dataset and performs exceptionally well for both BLEU and METEOR metrics.**

***Index Terms*—encoder-decoder, hyper-parameters, spatio-temporal, convolutional neural networks**

I. Introduction

A. Background

Convolutional neural networks (CNN) is a class of deep learning architecture inspired by the natural visual perception mechanism of the living creatures. It involves feed-forward artificial neural networks that has successfully been applied to analyzing visual imagery. A CNN consists of an input and an output layer, as well as multiple hidden layers [1]. CNNs are very effective in areas such as image classification and recognition.

Long Short Term Memory (LSTM) is a variant of Recurrent neural networks (RNN) that connects previous information with present training task. In the case of video description, LSTMs are useful because recalling the description of previous video frame could inform the understanding of the current video frame [2]. Original RNN is limited by the problem of long term dependencies where there is a long gap between relevant information or video frames as such current frames cannot gain from the information of previous frames. LSTM solves this problem by remembering long term information for a certain period of time.

B. Motivation

An encoder-decoder framework consists of two neural networks; the encoder and the decoder. Multiple frames are fed into the encoder to generate representations which are then fed into a decoder to generate the desired output. In terms of video to text, video clips serve as input to the encoder network, an intermediate representation is the output and input to the decoder whose output is a text - description of the events in the video. Encoder-decoder framework has been found to be useful in applications such as language translation and video to text frameworks. For the later, the encoder network is usually an image-trained CNN used on each frame separately. However, in the case of machine translation, it is natural to use a recurrent neural network (RNN) for the encoder as the input is a variable-length sequence of symbols[3]. While RNN, Gated Recurrent networks or variants of LSTM can be used as decoder in both cases since the output is a natural language sentence. Irrespective of the kind of neural network employed, there are variables known as **hyper-parameters** that determine the structure of the neural network. These parameters are usually set prior to training and optimizing weights and bias of the network. The values of these variables largely determine the performance of the final model. As such, it is imperative to understand the magnitude at which each hyperparameter affects the network.

C. Contributions

We investigate how hyperparameter configurations affects the performance of the encoder-decoder framework. We also introduce best configuration settings for deep neural networks frameworks for this task and track the intermediate text that gets generated throughout training

II. Related Work

CNN in video recognition has been implemented as a two-stream ConvNet architecture that incorporates spatial and temporal networks. For instance, in [3] ConvNet is trained on multi-frame dense optical flow and is able to achieve very good performance in spite of limited training data. This architecture is based on two separate recognition streams (spatial and temporal), which were combined by late fusion. The spatial stream performs action recognition from still video frames(images), while the temporal stream is trained to recognize action from motion in the form of dense optical flow. Both streams are implemented as ConvNets. This architecture is related to the two-streams hypothesis of the human visual cortex which has two pathways: the ventral stream (which performs object

recognition) and the dorsal stream (which recognizes motion). The input to the model is formed by stacking optical flow displacement fields between several consecutive frames. [4]–[6] implemented video description generators by constraining the domain of videos as well as the activities and objects embedded in the video clips. They also rely on hand-crafted visual representations of the video, to which template-based or shallow statistical machine translation approaches were applied. In our case, the video description generated is open-ended. In [7], hyerparameters that were discovered to play a vital in feed-forward deep neural networks were investigated. [8], performed exhaustive search on hyperparameter combinations that increased deep neural network's performance and compared it to the performance of random forest model for regression.

III. Methodology

The original implementation of [9] uses a certain set of configuration, which we explored. The aim of our methodology is to analyze how some critical hyper-parameters affect the performance of the network, in terms of the descriptions generated, as well as in terms of the evaluation metrics. Following are the hyperparameters we studied:

- nlayersattention: the number of layers used to compute the attention weights
- no of layers: number of layers used to compute logit
- number of hidden layers Init: the number of layers to initialize LSTM at time 0
- lstmencoder: if True, runs bidirectional LSTM
- prev2out: Feed previous word into logit
- ctx2out: Feed attention weighted ctx into logit
- patience: Number of epoch to wait before early stop if no progress
- maxepochs: The maximum number of epoch to run
- nwords: vocabulary size
- maxlen: maximum length of the description
- optimizer: SGD, adadelta, rmsprop, etc can be used
- batchsize: The batch size during training.

Looking at the above list of parameters, we decided to configure maxlen and lstmencoder. These two hyperparameters are very important in terms of generating the descriptions. One of the future scopes of the original implementation [9] was to check how different LSTM encoders could be integrated into the network. This proved to be a great reason for us to check out the effects different encoders could have on the results. Hence, we added the functionality for the LSTM Bidirectional encoder in addition to unidirectional LSTM. Uni directional LSTMs consider input only from the past. The hidden states are computed using only these past input. Uni directional LSTMs thus lack any context from the future. In contrast to this, bi directional LSTMs consist of two LSTMs, one which processes information from past to future and the other which processes information from future to past. Thus, combining two hidden states at any point in time, we get a context from both past and the future. The second hyperparameter we configured is the maximum length of the description. By configuring this parameter, we wanted to see whether concise true descriptions are generated by reducing the maximum allowed length of description or whether the model has a hard time to generate descriptions when the maximum length is reduced. Another reason for choosing this parameter for modification is that this parameter is very intuitive in its meaning and the results obtained can also be easily understood by everyone. That gives an additional advantage of being able to explain the results and effects easily.

We decided to compare model performance of different configurations by analyzing different evaluation metrics and checking the descriptions generated. The evaluation metrics we considered are as follow:

- BLEU: Scores are calculated for individual translated segments, generally sentences, by comparing them with a set of good quality reference translations. Those scores are then averaged over the whole corpus to reach an estimate of the translation's overall quality. Intelligibility or grammatical correctness are not taken into account. BLEUs output is always a number between 0 and 1. This value indicates how similar the candidate text is to the reference texts, with values closer to 1 representing more similar texts. [10]
- METEOR: The metric was designed to fix some of the problems found in the more popular BLEU metric, and also produce good correlation with human judgement at the sentence or segment level. This differs from the BLEU metric as it seeks correlation at the corpus level. [11]
- ROGUE: ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation. It is essentially of a set of metrics for evaluating automatic summarization of texts as well as machine translation. It works by comparing an automatically produced summary or translation against a set of reference summaries (typically human-produced). [12]
- CIDEr - a consensus-based evaluation protocol for image description evaluation. This protocol enables an objective comparison of machine generation approaches based on their human-likeness, without having to make arbitrary calls on weighing content, grammar, saliency, etc. with respect to each other. [13]

In order to visualize how video descriptions are generated after each epoch, we took a sample of description data from the training and validation set and displayed the ground truth (actual video description) along with the description generated by the network for the same samples in each epoch. In order to keep the same frame of reference, we tracked fixed descriptions in the train, text and validation sets. You can clearly see the progression in sophistication of the video-descriptions generated after each epoch. This is a good indicator of the result (apart from the metrics).

IV. Implementation

A. Data exploration

We explored a preprocessed Youtube2Text dataset - checked the structure, contents and what those contents represent. From

that we learnt that the entire dataset has 1970 video files. Each file has varying number of frames within it (depending on the length of the video clip). Each frame is represented using a 1024 dimensional frame-wise feature vector. This feature vector is obtained by selecting 26 equally-spaced frames out of the first 240 from each video and feeding them into the CNN.

B. Hyperparameter Configuration

Table I shows different hyperparameter configurations we tried and the corresponding cluster node the configuration was run on. For each configuration, we ran a batch file on each

TABLE I: Different Hyperparameter Configurations

Config No	LSTM Encoder	Max Length of Description	Node Name
1	lstm_uni	5	gtx1080
2	lstm_uni	12	gtx1080
3	lstm_uni	30	k20c
4	lstm_bi	5	k20c
5	lstm_bi	12	k20c
6	lstm_bi	30	gtxtitanx

of the nodes. We ran each configuration for only 50 epochs since each configuration takes about 20 hours to run and a node can be allocated on ARC for a maximum of 24 hours. After running each configuration, we wrote a python script which generated visualization of the metrics and also the descriptions.

C. Hardware architecture

1) Google Colab: We explored Googles Cloud Service for AI called Google Colab because it provides free GPU computation capability. It also supports shared development by allowing multiple people to work on the same ipython notebook, and supports multiple python versions as well. Thus, we decided to setup our environment and resolved dependencies on Google Colab workspace. However, the GPU provided by Google Colab is limited. First of all, the GPU functionality is not always available, and secondly executing commands on Google Drives workspace took a lot of time. A simple "rm -rf" took more than 5 minutes. Thus, we decided not to use Google Colab for our experiments.

2) ARC: A Root Cluster for Research into Scalable Computer Systems (ARC) [14] is a research project that addresses the challenges faced while testing software in a large-scale computing environment. It provides a mid-size computational infrastructure, called ARC (A Root Cluster), that directly supports research into scalability for system-level software solutions. Since we had to do experiments with the existing model, we needed the ability to at least load the data and run the model. ARC cluster provided us with that ability. The preprocessed Youtube2txt dataset is about 1.58 GB, therefore to load it in the memory and perform experiments on it we needed at least 5 GB of memory. Therefore, we are using the hardware (illustrated in Table II) provided by ARC for our exploration:

V. Result

A. Evaluation metrics

We analyzed the values of the key evaluation metrics by visualizing them using line charts. To do this, we have collected the values of these metrics on testing data (unknown data) after few epochs and plotted how these values vary over the epochs for each of the configuration. The X-axis of the images represents the increasing number of epochs, whereas the Y-axis represents the values of these parameters. Also, note that configuration 3 is the configuration of the original model and the remaining configurations are the experiments performed by our team to find out which alternate configuration, if any, could be comparable, or even better than the original one.

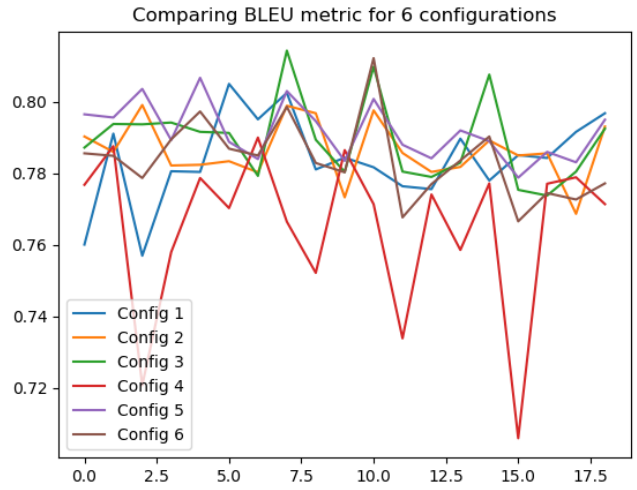

Fig. 1: Comparison between BLEU metric values

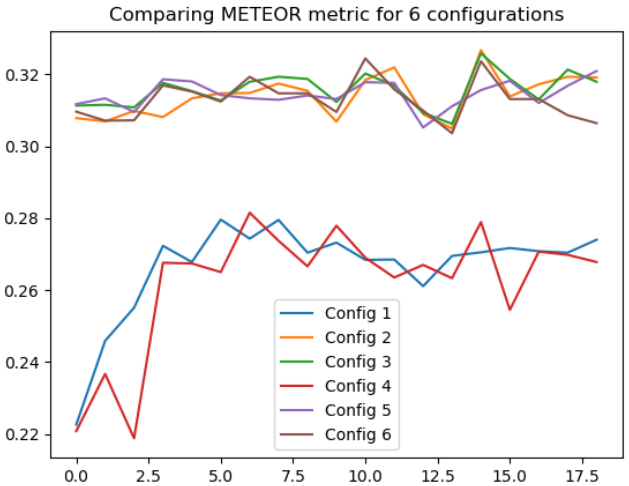

Fig. 2: Comparison between METEOR metric values

As we can see from figures 1, 2, 3 and 4, the red and the blue configurations (configurations 1 and 4) suffer distinctly in almost all parameters and rightly so! That is because both these configurations have the maximum length of descriptions equal to 5. Due to this restriction on the parameter, the truth

TABLE II: Hardware used for training model [15]

No	Hardware	Number of nodes	Memory Amount	Memory Bandwidth (GB/sec)	Stream Processors	Core Clock (MHz)	Memory Clock (MHz)
1	Tesla K20c Kepler	3	5.0GB	200	Stream Processors 2496	705	5200
2	Tesla K40c Kepler	1	12GB	288	Stream Processors 2880	875	3004
3	GeForce GTX Titan X Maxwell	3	12GB	336	Stream Processors 3072	1000	7010
4	GeForce GTX 1080 Pasca	2	8GB	320	Stream Processors 2560	1607	10,000
5	GeForce Titan X 10 series Pascal	1	12GB	480	Stream Processors 3584	1417	10,000

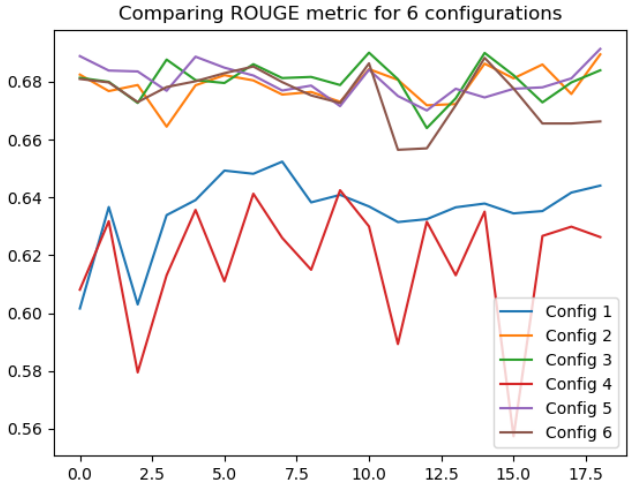

Fig. 3: Comparison between ROUGE metric values

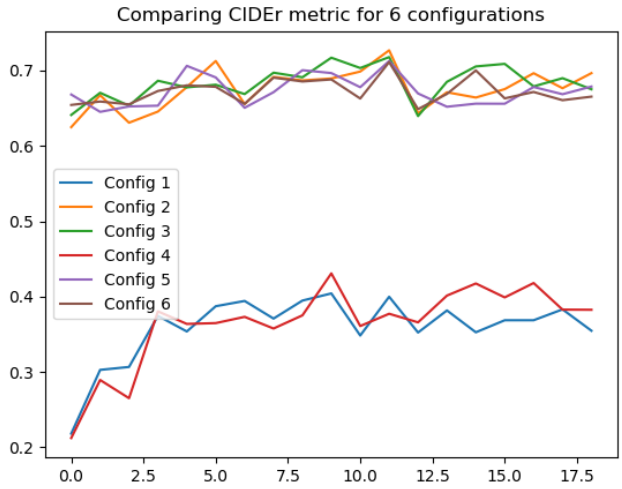

Fig. 4: Comparison between CIDEr metric values

samples selected by the model from the training set have very limited context and also the number of training samples from the data-set with satisfy max_len = 5 are less. This could have led to under-fitting of the model. On the other hand, the green and purple lines (configurations 3 and 5) seem to have similar and great scores for these metrics when derived from test data. However, configuration 5 seems to give consistently good values for all metrics, over the duration of all epochs, and therefore seems to give the best results from amongst the experimented configurations. Configuration 3, on the other hand, gives spikes of good values of these metrics on certain epochs but does not give consistent results.

B. Intermediate descriptions

In order to visualize the output of the RNN layer, after every epoch we generated and stored the intermediate descriptions of predetermined videos from training and validation data. This enabled us to compare the descriptions being generated as the model was being trained. Few of these descriptions from validation data are shown in figure 5. It can be observed that the descriptions at epoch 0 in both configurations are meaningless, since the initial weights assigned by the neural networks are arbitrary. However, as the number of epochs increase, the quality of descriptions gets better. This can specifically be observed by tracking the first truth statement of configuration 3. In epoch 0, it's blank. In epoch 30 it's "a cat is playing" and in epoch 50 it's "a cat is playing with a cat". The description generated by epoch 50 is the most similar to the truth i.e. "two cats playfully fight". Similar results can be observed for configuration 5.

VI. CONCLUSION

From the above results, we make the following conclusions:

- The line graphs of the values of metrics BLEU, METEOR, ROGUE, CIDEr are generated from the test values of these metrics. Config 3 gives spikes of good results in the graphs which is actually not an ideal model as its results will vary depending on the number of epochs. In contrast, Config 5 gives consistently good values of metrics on test data which shows that it is a robust model which is able to produce good descriptions for unseen data. Also, we can see that Configs 1, 4 are clearly not the ideal configurations for the model. Thus, we can conclude that the parameter max_len has an important role to play in deciding the quality of the model. Also, using bidirectional LSTM gives consistent results throughout the epochs because of its ability to processes information from past to future as well as from future to past, thereby giving a better context as compared to unidirectional LSTM.
- The intermediate text descriptions generated from the validation data provides a measure to intuitively understand the attention weights updated by the RNN layer. As the

```
------------- sampling from valid ----------
Truth  0 :  two cats playfully fight
Sample ( 0 )  0 :
Truth  1 :  the person is cutting tomato
Sample ( 0 )  1 :
Truth  2 :  a man is writing
Sample ( 0 )  2 :
Truth  3 :  a man is doing some skating stunts
Sample ( 0 )  3 :
Truth  4 :  the person is playing drums
Sample ( 0 )  4 :
Truth  5 :  someone petting a beaver
Sample ( 0 )  5 :
Truth  6 :  a person eating with hands and spoon
Sample ( 0 )  6 :
Truth  7 :  a boy trying to walk and dance
Sample ( 0 )  7 :
Truth  8 :  a man cuts some wood with a knife
Sample ( 0 )  8 :
Truth  9 :  a man wearing rubber gloves holds a hunting knife
perpendicular on a piece of wood and hammers the blade with a
wooden stick
Sample ( 0 )  9 :
```

(a) Config 3: Epoch 0

```
------------- sampling from valid ----------
Truth  0 :  two cats playfully fight
Sample ( 0 )  0 :  a cat is playing
Truth  1 :  the person is cutting tomato
Sample ( 0 )  1 :  a man is slicing a tomato
Truth  2 :  a man is writing
Sample ( 0 )  2 :  a man is playing with a paper
Truth  3 :  a man is doing some skating stunts
Sample ( 0 )  3 :  a man is playing with a dog
Truth  4 :  the person is playing drums
Sample ( 0 )  4 :  a man is playing drums
Truth  5 :  someone petting a beaver
Sample ( 0 )  5 :  a cat is playing
Truth  6 :  a person eating with hands and spoon
Sample ( 0 )  6 :  a man is holding a man
Truth  7 :  a boy trying to walk and dance
Sample ( 0 )  7 :  a woman is dancing
Truth  8 :  a man cuts some wood with a knife
Sample ( 0 )  8 :  a man is doing some tricks
Truth  9 :  a man wearing rubber gloves holds a hunting knife
perpendicular on a piece of wood and hammers the blade with a wooden
stick
Sample ( 0 )  9 :  a man is cutting a board
```

(b) Config 3: Epoch 30

```
------------- sampling from valid ----------
Truth  0 :  two cats playfully fight
Sample ( 0 )  0 :  a cat is playing with a cat
Truth  1 :  the person is cutting tomato
Sample ( 0 )  1 :  a man is cooking
Truth  2 :  a man is writing
Sample ( 0 )  2 :  a person is playing
Truth  3 :  a man is doing some skating stunts
Sample ( 0 )  3 :  a boy is running
Truth  4 :  the person is playing drums
Sample ( 0 )  4 :  a man is playing guitar
Truth  5 :  someone petting a beaver
Sample ( 0 )  5 :  a small animal is eating
Truth  6 :  a person eating with hands and spoon
Sample ( 0 )  6 :  a man is talking
Truth  7 :  a boy trying to walk and dance
Sample ( 0 )  7 :  a woman is dancing
Truth  8 :  a man cuts some wood with a knife
Sample ( 0 )  8 :  the person is doing the something
Truth  9 :  a man wearing rubber gloves holds a hunting knife
perpendicular on a piece of wood and hammers the blade with a wooden stick
Sample ( 0 )  9 :  a man is cutting a vegetable
```

(c) Config 3: Epoch 50

```
------------- sampling from valid ----------
Truth  0 :  two cats playfully fight
Sample ( 0 )  0 :  a a a a a a a a a a a a a a a a a a a a a a a a
a a a a a a
Truth  1 :  the person is cutting tomato
Sample ( 0 )  1 :  a a a a a a a a a a a a a a a a a a a a a a a a
a a a a a a
Truth  2 :  a man is writing
Sample ( 0 )  2 :  a a a is a a a a a a a a a a a a a a a a a a a a
a a a a a a
Truth  3 :  a man is doing some skating stunts
Sample ( 0 )  3 :  a a a a a a a a a a a a a a a a a a a a a a a a
a a a a a a
Truth  4 :  the person is playing drums
Sample ( 0 )  4 :  a a a a a a a a a a a a a a a a a a a a a a a a
a a a a a a
Truth  5 :  someone petting a beaver
Sample ( 0 )  5 :  a a a is a a a a a a a a a a a a a a a a a a a a
a a a a a a
Truth  6 :  a person eating with hands and spoon
Sample ( 0 )  6 :  a a a is is is is is a a a a a a a a a a a a a a
a a a a a a a a
Truth  7 :  a boy trying to walk and dance
Sample ( 0 )  7 :  a a a is a a a a a a a a a a a a a a a a a a a a
a a a a a a
Truth  8 :  a man cuts some wood with a knife
Sample ( 0 )  8 :  a a a is a a a a a a a a a a a a a a a a a a a a
a a a a a a
Truth  9 :  a cat jumping out a box
Sample ( 0 )  9 :  a a a is a a a a a a a a a a a a a a a a a a a a
a a a a a a
```

(d) Config 5: Epoch 0

```
------------- sampling from valid ----------
Truth  0 :  two cats playfully fight
Sample ( 0 )  0 :  a cat is playing
Truth  1 :  the person is cutting tomato
Sample ( 0 )  1 :  a man cooking his kichen
Truth  2 :  a man is writing
Sample ( 0 )  2 :  a man is playing a computer
Truth  3 :  a man is doing some skating stunts
Sample ( 0 )  3 :  a boy is playing with dog
Truth  4 :  the person is playing drums
Sample ( 0 )  4 :  a man is playing guitar
Truth  5 :  someone petting a beaver
Sample ( 0 )  5 :  a rabbit is eating
Truth  6 :  a person eating with hands and spoon
Sample ( 0 )  6 :  a man is talking
Truth  7 :  a boy trying to walk and dance
Sample ( 0 )  7 :  a girl is dancing
Truth  8 :  a man cuts some wood with a knife
Sample ( 0 )  8 :  a man is doing something
Truth  9 :  a cat jumping out a box
Sample ( 0 )  9 :  a cat is playing
```

(e) Config 5: Epoch 30

```
------------- sampling from valid ----------
Truth  0 :  two cats playfully fight
Sample ( 0 )  0 :  a cat is playing
Truth  1 :  the person is cutting tomato
Sample ( 0 )  1 :  a man cooking his kichen
Truth  2 :  a man is writing
Sample ( 0 )  2 :  a man is playing a computer
Truth  3 :  a man is doing some skating stunts
Sample ( 0 )  3 :  a boy is playing with dog
Truth  4 :  the person is playing drums
Sample ( 0 )  4 :  a man is playing guitar
Truth  5 :  someone petting a beaver
Sample ( 0 )  5 :  a rabbit is eating
Truth  6 :  a person eating with hands and spoon
Sample ( 0 )  6 :  a man is talking
Truth  7 :  a boy trying to walk and dance
Sample ( 0 )  7 :  a girl is dancing
Truth  8 :  a man cuts some wood with a knife
Sample ( 0 )  8 :  a man is doing something
Truth  9 :  a cat jumping out a box
Sample ( 0 )  9 :  a cat is playing
```

(f) Config 5: Epoch 50

Fig. 5: Intermediate descriptions generated by different configurations

number of epochs increase, the quality of description improves.

VII. Repository

The link to the GitHub repository is - https://github.ncsu.edu/rtnaik/video-description-cnn-rnn. For the dataset, refer https://goo.gl/cQDXNK

References

[1] Convolutional neural network. (2018, April 11). Retrieved from https://en.wikipedia.org/wiki/cnn

[2] C. (2015, August 27). Understanding LSTM Networks. Retrieved from http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[3] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. NIPS, 2014.

[4] A. Kojima, T. Tamura, and K. Fukunaga. Natural language description of human activities from video images based on concept hierarchy of actions. IJCV, 2002.

[5] A. Barbu, A. Bridge, Z. Burchill, D. Coroian, S. Dickin- son, S. Fidler, A. Michaux, S. Mussman, S. Narayanaswamy, D. Salvi, et al. Video in sentences out. UAI, 2012.

[6] M. Rohrbach, W. Qiu, I. Titov, S. Thater, M. Pinkal, and B. Schiele. Translating video content to natural language descriptions. In ICCV, 2013.

[7] Alexios Koutsoukas Keith J. Monaghan Xiaoli Li 0013 Jun Huan Deep-learning: investigating deep neural networks hyper-parameters and comparison of performance to shallow methods for modeling bioactivity data. 42:1-42:13 2017 9 J. Cheminformatics 1 https://doi.org/10.1186/s13321-017-0226-y db/journals/jcheminf/jcheminf9.htmlKoutsoukasMLH17

[8] Ma J, Sheridan RP, Liaw A, Dahl GE, Svetnik V (2015) Deep neural nets as a method for quantitative structureactivity relationships. J Chem Inf Model 55:263274

[9] Yao, Li, et al. "Describing videos by exploiting temporal structure." Proceedings of the IEEE international conference on computer vision. 2015.

[10] BLEU. (2018, April 19) Retrieved from https://en.wikipedia.org/wiki/BLEU

[11] METEOR. (2018, April 19) Retrieved from http://www.cs.cmu.edu/ alavie/METEOR

[12] What Is ROUGE And How It Works For Evaluation Of Summarization Tasks? (2018, April 19) Retrieved from http://rxnlp.com/how-rouge-works-for-evaluation-of-summarization-tasks/#.Wtl4GdPwb-Z

[13] Vedantam, Ramakrishna, C. Lawrence Zitnick, and Devi Parikh. "Cider: Consensus-based image description evaluation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

[14] ARC (2018, April 20) Retrieved from http://moss.csc.ncsu.edu/ mueller/arc.html

[15] NVIDIA hardware (2018, April 20) Retrieved from http://moss.csc.ncsu.edu/ mueller/cluster/nvidia/

VIII. Honor Code

By submitting this project report and code, we agree that we haven't copied code from anywhere else (including our own previous projects). Any code referred has been appropriately cited. We agree that, unless cited, every sentence that went into the report is written by the group members and not copied from anywhere else.

IX. Individual Project Member Contributions

Table III shows each task and its description with the corresponding time taken and the group members that worked on it.

TABLE III: Task-wise contribution

No	Task	Description	Time taken (days)	Members
1	Literature Review	Studying papers cited by our base paper, understanding the recent trends and related work	10	Ruth, Nirav, Udit, Rohit
2	Studying Existing Architecture	Understanding the existing encoder-decoder architecture as presented in our base paper. Studied CNN, RNN and LSTMs thoroughly.	20	Udit, Nirav, Rohit, Ruth
3	Defining each Hyper-parameter	Studying how other research works tune hyper-parameters to enhance performance of deep neural networks and deriving the meaning of the hyper-parameters	5	Ruth
4	Research about Configuration Parameters	Analyze the current hyper-parameter configuration and identify the important ones for experimentation	13	Udit, Rohit
5	Deciding configuration parameters of interest	With the help of the background about the parameters, finalizing the parameters for experimentation	4	Udit, Nirav
6	Report Writing	Abstract, Introduction, and Related Work	5	Ruth
7	Understanding metrics for quality of text summarization	Understanding various evaluation metrics like BLEU, METEOR, ROGUE, CIDEr	5	Rohit, Nirav
8	Designing experiments	Various hyperparameter configurations to be tested. Created multiple workspaces in order to run them in parallel	6	Rohit, Udit
9	Experiments on Google Colab	Did an environment setup on Google Colab and ran multiple configurations on it to leverage GPU support	13	Nirav, Udit
10	Environment set up on ARC Cluster	Installed dependencies on ARC via source and wrote a shell script for the same. Also added support for having multiple environments setup in same user workspace to provide capability to train multiple models in parallel	25	Nirav, Rohit, Udit
11	Running batch jobs on ARC Cluster	Coordinated with ARC users to understand batching jobs and added that support in our project by creating multiple batch files for different configurations and executing them by a simple script	15	Nirav, Rohit, Udit
12	Executing experiments and preserving results	Modifying the required files in order to run the different configurations and preserving the results before executing running another configuration	7	Udit, Rohit
13	Tracking intermediate video descriptions	Generated and tracked intermediate video descriptions after each epoch while training the model. Debugged the code to first get deserialized video files and then used those video ids to predefine a list of video ids to track while training	4	Nirav, Udit
14	Script for generating visualization of metric values	Python script for parsing the results file and visualizing metrics for different configurations	5	Rohit, Nirav