Machine Learning 2 Lydia Teinfalt
Spring 22 04/25/2022

1 Introduction

An image without a caption can sometimes lead viewer to get the wrong impression of what is going on in a picture. A good caption can provide the viewer with necessary information, focus on key details, and enhance the overall understanding of the image by using just a few words. This project demonstrates machines' capability to generate captions for pictures in natural language sentence format by using a machine learning model. The model using TensorFlow based is built on the architecture described in the Show, Attend, and Tell Paper. We evaluate the model using BLUE score that compares the official caption with the automated generated caption.

2 Description of Individual Work

As a group, we created a project schedule and outline with our task to complete our project. I kept our project schedule on track and lead group meetings. I create a Teams site so we could organize our project work. In terms of my work towards the project, my work for the project focused on the following tasks:

- Read and grasp major concepts from Show, Attend, and Tell Paper
  a. Researched Long-Short Term Memory (LSTM)
  b. Obtained high level understanding of LSTM with visual attention
- Explored COCO 2014 dataset
- Explored TensorFlow or PyTorch frameworks to implement the model
- Familiarized with TensorFlow 2.0's tutorial of for image captioning
- Implemented the model in the tutorial in PyCharm as a jumping off point for enhancements
- Made the following contributions to the code:
  a. Replaced InceptionV3 with VGG Network
  b. Researched VGG network
  c. Experimented with building translation model using NMT
  d. Worked on integrating Google Translate API

3 Details of Individual Work

We divided our group project based on our individual strengths. Alice's undergraduate is in mathematics, Adel's major is data analytics, and my background is in coding, so it made sense that my focus was on implementing image captioning model in TensorFlow. We used the notebook offered by TensorFlow tutorial in Image Captioning as a jumping off point. Our initial class is called image_captioning.py. The following modifications were made to class:

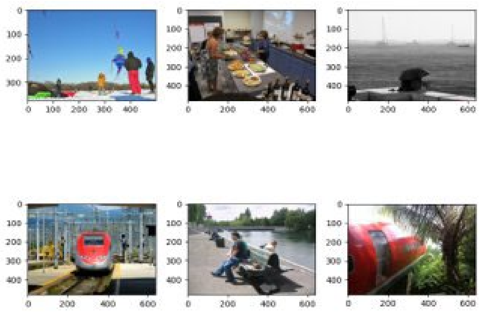
- Selecting 10,000 images instead of 6,000 training images from COCO dataset (2014)

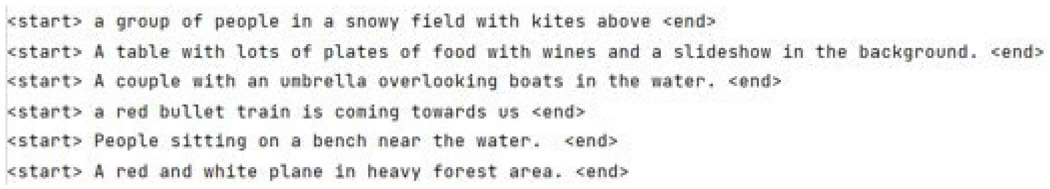
```
# Select the first 10000 image_paths from the shuffled set.
# Approximately each image id has 5 captions associated with it, so
that will
# lead to 30,000 examples.
train_image_paths = image_paths[:10000]
```

- Created a figure to plot a few random sample data as part of exploratory data analysis:

```
# Added this to plot examples from the dataset
fig = plt.figure(figsize=(8,8))
i = 1
for j in range(6):
  n = np.random.randint(0, 100)
  ax1 = fig.add_subplot(2,3,i)
  plt.imshow(Image.open(img_name_vector[n]))
```

```
        print(train_captions[n])
        i += 1
        plt.savefig("sample_data.pdf")
```





```
<start> a group of people in a snowy field with kites above <end>
<start> A table with lots of plates of food with wines and a slideshow in the background. <end>
<start> A couple with an umbrella overlooking boats in the water. <end>
<start> a red bullet train is coming towards us <end>
<start> People sitting on a bench near the water.  <end>
<start> A red and white plane in heavy forest area. <end>
```

- Implemented VGG16 network instead of InceptionV3

```
def load_image(image_path):
  img = tf.io.read_file(image_path)
  img = tf.io.decode_jpeg(img, channels=3)
  img = tf.keras.layers.Resizing(224, 224)(img)
  img = tf.keras.applications.vgg16.preprocess_input(img)
  return img, image_path
```

```
image_model = tf.keras.applications.VGG16(include_top=False,
                                          weights='imagenet')
features_shape = 512
attention_features_shape = 49
```

- Use google translate library to translate English caption to German (or any language supported)

```
sentence = ' '.join(result)
translated_captions = translator.translate(sentence, dest= lang)
print(sentence, ' -> ', translated_captions.text)
```

- Use model to predict captions for New Yorker cartoons

```
image_url0 = 'https://raw.githubusercontent.com/nextml/caption-contest-
data/gh-pages/cartoons/667.jpg'
image_url1 = 'https://raw.githubusercontent.com/nextml/caption-contest-
data/gh-pages/cartoons/671.jpg'
image_url2 = 'https://raw.githubusercontent.com/nextml/caption-contest-
data/gh-pages/cartoons/670.jpg'
image_list = [image_url0, image_url1, image_url2]

for image_url in image_list:
    image_extension = image_url[-4:]
    image_path = tf.keras.utils.get_file('image'+image_extension,
origin=image_url)
    result, attention_plot = evaluate(image_path)
    plot_attention(image_path, result, attention_plot)
    # opening the image
    Image.open(image_path)
```

4 Experiments

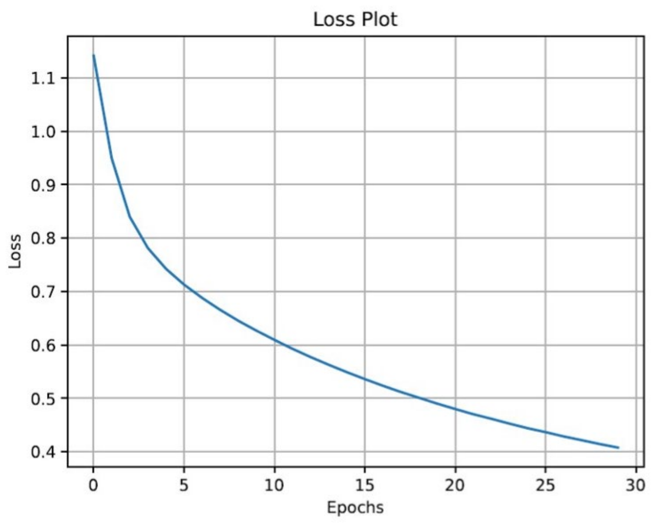
We assume that training with 10000 values versus 6000 from COCO 2014 dataset would improve our model. Using the BLEU scores, we saw minimal improvements with the larger dataset. It does not seem to be worthwhile to increase the dataset because it does not yield much higher BLEUR scores and instead increases our complexity with number of parameters and slows the performance of training the model.

Dataset Size	BLEU-1	BLEU-2	BLEU-3	BLEU-4
~40,000 Train Images	44.5	23.3	12.2	6.5
~24,000 Train Images	44.3	22.0	10.8	5.7

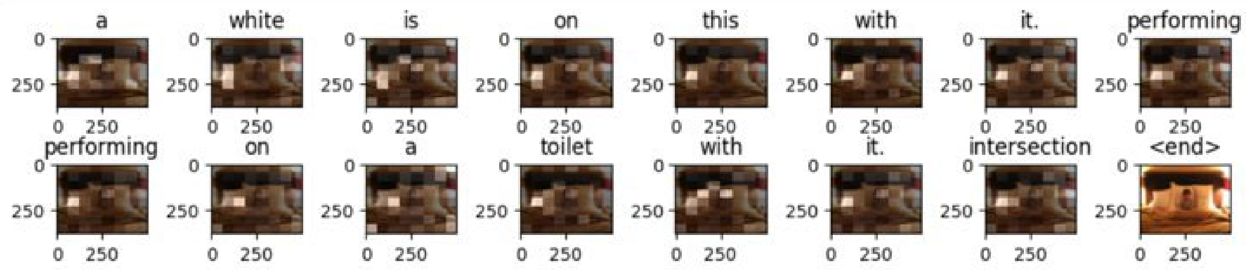
We trained the model by varying the number of epochs. The lowest number of epochs being 20 and the highest being 100. By incrementally increasing the number of epochs to train the model, the loss plot shows a steady decrease towards 0. The BLEU score seems to worsen with the increased number of epochs. The best score was a result of training on 10 epochs.
The model's performance is slow to train. On average, an EC2 server on AWS cloud took approximately 160 seconds (about 2 and a half minutes) per epoch to train the model. Assuming that is the average time to train the model, it would take approximately 266.7 hours (about 1 and a half weeks) to train the entire model if we had 100 epochs. The model performed better on a GPU deployed in Google Cloud Platform – approximately 90 seconds (about 1 and a half minutes) per epoch. On a GCP GPU server, it would take us only 150 hours (about 6 and a half days) to train the model if it had 100 epochs.

5 Results

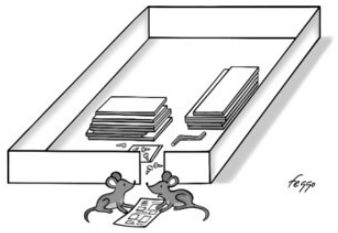
We trained the model on a subset of the COCO 2014 dataset. This resulted in 40,018 training images and 200,090 training captions because there are five captions per image. For test, we have 10,003 images and 50,015captions. We trained using 50, 65, 85, and 100 epochs. Here is the loss plot when we trained using 25 epochs using Adam optimizer. The plot show a steady decline in loss and approaching 0.4.



The model generated a picture from test dataset below. The plot shows where the area of the picture that the model is focusing on to generate the caption. It does not make sense why it identifies a toilet. From a sentence structure, it has the verb "performing" twice and ends the sentence with with "it." But then erroneously generate a single word "intersection" that is not a proper sentence.

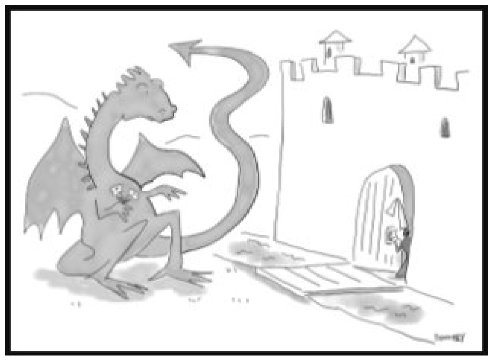


Then we used the trained model to predict captions for New Yorker cartoons. To be fair, we have been training on pictures so to switch to cartoons it has to be acknowledged that cartoons and pictures are <u>not</u> equivalent.



Generated Caption: A table [UNK] a table and a box placed on it. <end>
German Translation: Ein Tisch [UNK] ein Tisch und eine darauf gestellte Kiste. <Ende>

The model correctly identified rectangular shape as a box and unsure how a table was detected. It entirely missed the two mice in the caption but an artist rendering of a mouse is quite different from an actual mouse so that is understandable. The German translation is perfect even if the English sentence does not make sense. Note that [UNK] is for "unknown".



Generated Caption: Many cows lounging at sheep goat <end>
German Translation: Viele Kühe räkeln sich bei der Schafziege <Ende>
The model identified an animal in the image but incorrectly classified it as a cow. The dragon above also was misclassified as a sheep goat. In terms of the caption the word lounging would make more sense to by the word "lunging" especially following with the word "at". The combination of words sheep goat could be interpreted as the model thinking that a sheep or goat is in the picture. The English caption translated to German is well done except for the sheep goat which is an unknown breed.

6 Summary and Conclusions

We built a working image captioning model in TensorFlow using attention mechanism. The paper was state of the art in 2016 but is now probably obsolete. Our model performed poorly compared to the results reported in the Show, Attend, and Tell Paper. The visual attention mechanism is still a fascinating discovery in that it resembles what humans do on an intuitive level. There is a lot of areas of improvement for the model. To improve, we could the model using the entire COCO2014 dataset instead of only a subset. In addition, we could parameterize the language selection for translating the image caption so that the output can be in any language that the user specifies. We could have used another image dataset like Flicker8k and create a loop to iterate over the images and generate captions.
Total lines of code = 333
Total lines of code modified = 7
Total lines of code added = 17
Percentage of code = (333-7)/ (333+15) * 100 =93.7%

7 References

COCO - Common Objects in Context. (n.d.). COCODataset. Retrieved April 11, 2022, from https://cocodataset.org/

Image captioning with visual attention | TensorFlow Core. (2022, January 26). TensorFlow. Retrieved April 15, 2022, from https://www.tensorflow.org/tutorials/text/image_captioning

Section 508 (Federal Electronic and Information Technology). (n.d.). U.S. Access Board. Retrieved April 25, 2022, from https://www.access-board.gov/law/ra.html

Show, Attend And Tell - Paper Explained. (2021, May 3). YouTube Halfling Wizard. Retrieved April 16, 2022, from https://www.youtube.com/watch?v=y1S3Ri7myMg&t=6s

Xu, Ba, Kiros, K. J. R. (2016, April 19). *Show, Attend and Tell: Neural Image Caption Generation with Visual Attention*. Arxiv.Org. Retrieved April 4, 2022, from https://arxiv.org/pdf/1502.03044.pdf?msclkid=97863852c49311ecb15903abb2f078b2