Novel Image Captioning

Daniel Thirman
CS 224D
DThirman@Stanford.edu

Abstract

In this paper I describe a model which is used to generate novel image captions for a previously unseen image by using a combination of a recurrent neural network and a convolutional neural network. This model is trained on the flickr30k and MS-COCO datasets of images and captions and scores a perplexity that is comparable to that of state of the art implementations. The network is evaluated by computing its perplexity as a function of how well the language model scores the sentence and how likely the sentence is given the image. In this paper, I present three different implementation a baseline model that learns its own embeddings for the vocabulary along with a basic RNN, a model that uses the pre-trained GloVe word vectors, as well as model that uses the GloVe word vectors as well as using Gated Recurrent Units to solve the vanishing gradient problem in the network.

1 Introduction

There are many applications for being able to automatically obtain a description for a picture. Some of these include image recognition for autonomous robotic systems which need to be able to recognize and process what they see, creating an image database that is search-able by keywords without having to manually tag and describe the different images that are added to the database, as well as many other possible applications. These applications are becoming increasingly relevant as technology processes so being able to determine the sentence level description of an image is a very important and interesting task. In the past several years, there have been many attempts by top researchers in the field to solve the problem and they have implemented models that are able to produce very good results that create natural sentences that describe the content of the images very well. Many well known companies, such as Microsoft and Google, have published models recently that attempt to solve the problem. In this paper I create a model that is inspired by other researchers' work on the problem and compare my results to the state of the art results in the field.

There have been some attempts that rather then generating new captions to describe the image instead choose to try find the best caption in their database that matches to the image. While this method is guaranteed to express captions that our naturally written and clear to understand, it fails to describe images that have unique combinations of objects or objects presented in an unusual way. Novel image captions are captions that are generated by the model from a combination of the image features and a language model instead of matching to an existing captions. Generating novel image captions solves both of the problems of using existing captions and as such is a much more interesting and useful problem.

The model I chose to implement to solve this problem is a multimodal neural network composed of a convolutional neural network and a recurrent neural network. The CNN is used to determine the image features and the RNN is used to generate the language model. These two networks are then combined with a weighted sum and the result is then used to predict the next word in the sentence that is used to describe the image that is inputted. I implemented and ran tests on three different versions of this model. The first is the baseline model which uses a basic RNN and learns the vector embeddings for the vocabulary by itself, the second is a model that uses the pre-trained GloVe word vectors instead of learning them itself, and the final model continues to use the GloVe word vectors but also uses Gated Recurrent Units in the RNN to increase performance and alleviate the vanishing gradient problem. Each successive model scores a lower perplexity and produces better results.

The data set that is used to train this network is a combination of MS-COCO and the Flickr30k data sets. Both data sets contain large amount of images and captions that can be used to train and evaluate the network.

2 Related Work

This problem has been one of interest to many researchers in the past few years. There have been many attempts to solve the problem using a variety of approaches. The models that are implemented in this paper are inspired by many of the other work performed by other researchers in the field.

Recently, Microsoft has released the MS-COCO database for anyone to use to work on this problem and has set up and challenges and offered prizes to those who do well on the problem so there have been many people working on the same database obtaining strong results.

3 Approach

3.0.1 Baseline model

The model that I chose to implement for this project is a multimodal neural network that is composed of a convolutional neural network that is used to detect images features as well as a recurrent neural network, that is trained as a language model and predicts the next word given the context of the previous words and the image features supplied by the CNN.

For this project I chose focus on the RNN instead of on CNN, since the RNN is more relevant to natural language processing which is the topic of the project, as such I chose to make use of a pre-trained CNN that would be able to extract image features and I would not have to train one from scratch. The pre-trained CNN I chose to use was one called AlexNet. ALexNet is a state of the art CNN implementation that is used to predict the probabilities of whether classes of images are contained inside a given image. There are 1000 classes of objects and running AlexNet on an image will return a vector of size 1000 which is the probability of each class.

The output of the CNN is then fed into the RNN to predict the next output word of the caption. The RNN is a single layer network with a hidden state of length 512. Different lengths were tried for the size of the hidden state however 512 produced better results than other lengths. At each time step the next hidden state is computed using a combination of the previous state, the word vector for the input word, and the CNN output for the image that the caption is being generated for. The equation for the next hidden state is equal to:

$$State_i = \sigma(State_{i-1} * H + Word_i * I + Image * N) + b$$

Where H is (Hidden size X Hidden size),State is the hidden state of the RNN, I is (Word Embedding Size X Hidden Size), Word is the word vector obtained by looking up the current input word in embedding, N is (Size of CNN output X Hidden Size), and Image is the output from the CNN, and b is the bias. A drawing of one node in the RNN is found in figure 1. σ is sigmoid function which is used as the non-linearity of the RNN.

The output of each time step is then projected to predict the probability of each word appearing next in the sentence description of the word. The final projection into the vocab size is computed in a multimodal layer that computes a weighted the outputs of the projections of the RNN and the CNN. The equation for the projection at each time step is equal to:

$$output = A * (RNN_i * U) + (1 - A) * (CNN * V) + b$$

Where A is a variable learned by the network that ensures calculates how much of the RNN or CNN output should be included in the final projection. RNN_i is the output of the RNN at time

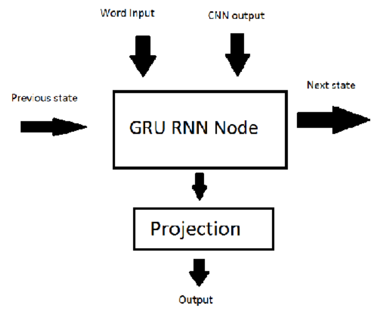

Figure 1: One node in the RNN

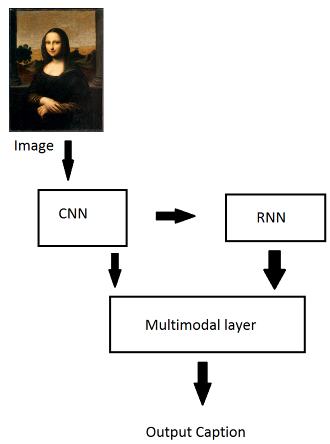

Figure 2: Entire multimodal neural network

step i, U is (Hidden Size X Vocab Size), CNN is the output of the CNN for the given image, V is (Size of CNN ouput X Vocab Size) and b is the bias.

This weighted sum of RNN and CNN output ensures that both the image features and the language model both factor in the final output probability at each time step and the best word for both inputs is most likely to be selected. The structure of the whole network is shown in figure 2.

3.0.2 GloVe Vector based model

The previous model did not generate particularly good results and while its captions might capture some of the semantic meaning of the image the language model was relatively poor. I reasoned that this was because although I had lots of examples in my data set most captions were relatively short so there was not enough text to adequately train the word vectors that were being used in the embedding of the vocab. So, I decided to use the pre-trained GloVe word vector. GloVe word vectors are word vectors that are trained based off the co-occurrences of word pairs in a very large corpus. Since the corpus used to train them was substantially larger than the corpus of sentences used in the image captions I had, using these captions would results in a better representation of the word similarities and would create a better language model.

3.0.3 Gated Recurrent Unit Model

While the previous model did do a better job than the baseline model of expressing captions it still had its problems. Since the captions generated were not of insignificant length, one problem that this network suffered from was the vanishing gradient problem. In the vanishing gradient problem the values of the gradients diminish rapidly as they are back-propagated to previous states and they have smaller and smaller changes on the variables in the states until the change is insignificant and the variables at states a few steps away do not train at all. One solution to this is to use a gated recurrent unit or a GRU. GRU's are a more complicated node structure than a base RNN in which the value of the previous state is scored for how important it is in the next state and only as much as is optimal is used to compute the next state. This helps alleviate the vanishing gradient problem as the values of the gradients are no longer exponentiated at each step in back propagation and the gradients no longer diminish so quickly. The new equation for the next hidden state is

$$z = \sigma(State_{i-1} * H_z + Word_i * I_z + Image * N_z) + b_z$$

$$r = \sigma(State_{i-1} * H_r + Word_i * I_r + Image * N_r) + b_r$$

$$\hat{h} = \tanh(r * State_{i-1} * H + Word_i * I + Image * N) + b$$

$$State_i = z * State_{i-1} + (1 - z) * \hat{h}$$

Here z is the update gate which is used to calculate how much of the new state should come from the previous state and how much of it should come from the new value that it is calculating, and r is the reset gate which calculates how much of the previous state should go into the new value that is being calculated, $\hat{h}$. All other values and matrices have the same values and dimensions as they did in the previous baseline model.

4 Experiment

4.1 Data set

The data set that I have used to train my model is a combination of the MS-COCO and Flickr30k data sets. MS-COCO is a publicly available data set of images and captions provided by Microsoft that is used to test and train network for exactly this purpose. The data set contains over 300 thousand images and along with each image contains a sentence level description of the image and a list of classes of objects that are contained in each image.

Flickr30k is a data set distributed by researchers at the University of Illinois. It contains 30 thousand images taken from the website Flickr and contains multiple captions for each image for a total of nearly 150,000 captions. By combining these two data sets I had nearly 500,000 captions and over 300,00 images to train the network on.

4.2 Evaluation

The model was scored by using the computing the perplexity based off the how well the sentence fit the language model and how well it worked to describe the image that it had been given to caption. The score of the language model was calculated based off of how similar successive word vectors were to each other. Since, word that occur more closely to each other are more likely to have a similar value since the vectors are created based off their co-occurences, so similar word vectors are more likely to have a low perplexity. The score for how it fits the image is created by calculating how likely it is for the word to be selected based off of the image. To accomplish this I went through the data set and calculated likely each class of object is given that a given word is included in its caption. Then to compute the perplexity based off how well each word fits the image, the likelihood of class given a caption word is compared to the likelihood of the classes given the image. The equation for calculating the loss is as follows:

$$J(\theta) = -\sum_{i=1}^{\|V\|} y_i^{(t)} * log(\hat{y_i}^{(t)} - \sum_{j=1}^{\|I\|} \sum_{i=1}^{\|V\|} P(y_i^{(t)} | \widehat{Image_j}) * log(\hat{y_i}^{(t)})$$

4.2.1 Results

The perplexity scores for each of the different models is shown in the following table.

Figure 3: Example images 1,2, and 3

Model	Train Perp.	Valid Perp.	Test Perp.
Baseline	65.23	11.18	8.3925
GloVe Model	152.63	70.23r	32.93
GloVe + GRU 136.16	50.65	26.42	

While the scores for the baseline model are initially relatively high and not particularly good, this is to be expected as the model had many problems such as not being able to train its own word vectors sufficiently well and falling victim to vanishing gradient problem. However, the successive models each do better than the previous versions and the final results are relatively good.

Current state of the art implementations scored and trained on the MS-COCO databases have scored a perplexity of 14.23. My value is comparable to that value and while mine is a little higher that is to be expected as I have had a much smaller amount of time and computing power and there were a decent amount concessions I had to make in the interest of what was possible in the given time given my resources. Examples In this section I will go through a few captions generated by my models and point out the parts that my model does well at and the parts that it does poorly on. Here are some captions created by the GloVe and GRU models, I chose not to include the baseline captions as they are worse than the captions produced by these and rather hard to understand.

GloVe Based Model

1. round in a young girl in the background a women on a large red pour a boy and woman a restaurant the preparing a girl with a circle in man is relaxing a man hat

2. they are colored dressed look at another person small group of people bending down a soccer ball wtwo people are gathered on the open in its forefront

3. two kids playing his tan room guy in pink is one a crd bucket a man wearing a bench with front of a sheath in pat of others and man holding a counter
GRU Model

1. a woman is wearing a black shirt and woman in a red dress shirt and a woman are working on a restaurant a young woman holds a fruit on a black of a woman at a table with a wooden of woman

2. boy in white are practicing walking down in street with many children in yellow uniforms four men and a yellow belt belt boys are playing karate on a line a competition player in a field soccer children play

3. sits in front of a building a man in jeans and sunglasses is standing a girl in a stands in the middle of the street a young woman is standing he is looking at a woman in a city street an older woman is sitting on a sidewalk

Looking at the captions it can be seen that while the sentences can be somewhat awkward in some places they still do a good job representing what is going on in images, while still making some mistakes. Also, it can be seen how the results from the GRU based network are better than the results than the simpler model in many cases. For example, in image 1 both models are able to reflect that there women in the picture the GRU model is able to give more details such as that there is fruit rather than just that there is something round as simpler model suggests, or that it is get that the people in the third image are out in a sidewalk and not inside a room with tan walls. Both models seem to do a good job getting the colors that are in the image though, by recognizing the color clothes that the people are wearing this can be seen by getting the woman in red in image 1, the tan walls in image 2, and the white and yellow uniforms in image 3. However, both networks still make some mistakes like how the GRU network sees a man in sunglasses and jeans even though there is no one dressed like that. So, while there is still some mistakes being made even the state of the art model makes lots of mistakes, so this model can still be seen as relative success as it does do a sufficient job captioning and expressing what is going in images.

5 Future Work

One of the main improvements that could be done for this model would be to train the CNN along with RNN. Since I did not have the time or computing power to back propagate the errors to the CNN I used the pre-trained AlexNet network. While that does produce good results the network definitely perform better if it could train the CNN at the same time and the errors would affect the values in the CNN.

Another area for improvement would be to try more complicated ways of scoring the language model to get better results. If the model was switched to scoring the words in caption based on a window it would probably significantly help make the language model more robust and help make the end result better and more natural to read and understand.

After that I would try to use a more complicated RNN structure. A multi-layer RNN may be able to better learn the features of the image and the language model and perform better results.

References

[1] Mao, Junhua, Xu, Wei, Yang, Yi, Wang, Jiang, and Yuille, Alan. Deep captioning with multimodal recurrent neural networks (m-rnn). arXiv:1412.6632, December 2014.

[2] Vinyals, Oriol, Toshev, Alexander, Bengio, Samy, and Erhan, Dumitru. Show and tell: A neural image caption generator. arXiv:1411.4555, November 2014.
[3] Karpathy, Andrej and Li, Fei-Fei. Deep visual-semantic alignments for generating image descriptions. arXiv:1412.2306, December 2014.
[4] Chen, Xinlei, and Zitnick, C Lawrence. Learning a Recurrent Visual Representation for Image Caption Generation. arXiv:1411.5654, November 2014.
[5] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation
[6] Peter Young, Alice Lai, Micah Hodosh and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions, Transactions of the Association for Computational Linguistics, 2(Feb):67-78, 2014. **15**(7):5249-5262.