Image Captioning with Attention

Blaine Rister (blaine@stanford.edu), Dieterich Lawson (jdlawson@stanford.edu)

1. Introduction

In the past few years, neural networks have fueled dramatic advances in image classification. Emboldened, researchers are looking for more challenging applications for computer vision and artificial intelligence systems. They seek not only to assign numerical labels to input data, but to describe the world in human terms. Image and video captioning is among the most popular applications in this trend toward more intelligent computing systems.

For our course project, we designing an imagecaptioning system using recurrent neural networks (RNNs) and attention models. Image captioning is the task of generating text descriptions of images. This is a quickly-growing research area in computer vision, suggesting more "intelligence" of the machine than mere classification or detection.

RNNs are variants of the neural network paradigm that make predictions over sequences of inputs. For each sequence element, outputs from previous elements are used as inputs, in combination with new sequence data. This gives the networks a sort of "memory" which might make captions more informative and context-aware.

RNNs tend to be computationally expensive to train and evaluate, so in practice memory is limited to just a few elements. Attention models help address this problem by selecting the most relevant elements from a larger bank of input data. Such schemes are called "attention models" by analogy to the biological phenomenon of focusing attention on a small fraction of the visual scene.

In this work, we develop a system which extracts features from images using a convolutional neural network (CNN), combines the features with an attention model, and generates captions with an RNN.

2. Prior Work

There has been much prior work on attention models and image captioning with neural networks. In this class alone, at least two of the instructors have written articles on the topic. Karpathy and Li generated captions for distinct locations with an image [2]. Yeung et al. used attention models to classify human actions from video sequences [10]. Many researchers, such as Xu et al., have used attention models to visualize and interpret which parts of the scene are most important to a network's determination of words [9]. Even this cursory examination reveals a wide range of attention models and captioning architectures. Many other architectures can be found in the literature [7, 5, 8].

3. Architecture

Our model will has three main parts: a convolutional neural network (CNN) that extracts features from images, an attention mechanism that weights the image features, and an RNN that generates captions to describe the weighted image features. In the following sections, we will describe each in turn.

3.1. Convolutional Neural Network

A convolutional neural network is a standard feedforward neural network, except that instead of computing the affine function $f(x) = Ax + b$, we restrict A to be a Toeplitz matrix. This is the same as saying that A is a convolution. This basic idea generalizes to images, where we have a 2D convolution over each of the K color channels. Each layer performs an affine operator consisting of convolutions and element-wise summations:

$$f(x) = \sum_{k=1}^{K} g_k * x + b_k.$$

In the context of neural networks, we often follow these affine "convolutional layers" with a "nonlinearitie." In fact f is an affine operator on x, and nonlinear itself. The most commonly used nonlinearity is the rectified linear unit (ReLU), which is nothing more than the hinge loss

$$r(x) = \max(0, x)$$

where the maximum is taken element-wise. Thus the total unit of a convolutional neural network is

$$h(x) = \max\left(0, \sum_{k=1}^{K} g_k * x + b_k\right). \quad (1)$$

In this work, we have chosen to use a pre-existing CNN model, called ResNet-50. This model consists of 50 layers each performing a similar operation to 1. The subtle differences between ResNet-50 and other CNNs are beyond the scope of this paper.

3.2. Recurrent neural networks

While CNNs are adept at signal processing, they cannot easily express the idea of pattens in sequences. Recurrent neural networks (RNNs) are similar to feed-forward neural networks, except their outputs are fed back into the input. The networks maintain a "hidden state" that allows them to adjust their behavior throughout iterations of this feedback loop. RNNs are considered state of the art in machine translation and other tasks involving generating text, as they easily learn the patterns of human grammar.

In this work, we leverage a powerful variant on the RNN idea called a Long Short-Term Memory (LSTM) unit. In this paper, we will simply state the equations for LSTMs, and not delve into their interpretation. Given a state at time t, the LSTM update equations are

$$\begin{aligned}
\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} &= \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} \circ W \begin{pmatrix} h_{t-1} \\ x \end{pmatrix} \\
c_t &= f \circ c_{t-1} + i \circ g \\
h_t &= o \circ \tanh(c_t)
\end{aligned}$$

where σ denotes the sigmoid function, and tanh denotes the hyperbolic tangent. The vectors i, f, o, g are called "gates" of the LSTM for reasons that are beyond the scope of this paper.

In image captioning, the input x is a vector representing a specific n-gram, which is called an "embedding." We learn the parameters of this embedding as an affine transfomation of 1-hot vectors. After processing, the hidden state h is transformed to vector of class scores, where each n-gram, which in our case is just a word, corresponds to a distinct class. Thus, the predicted word at time t w_t is just

$$w_t = \arg\max_i (Ah_t + b)_i$$

where A, b are learned parameters. Note that we initialize the hidden state h_0 with the image features. From here, it is clear that this is just a standard classification problem, and can be trained with the softmax loss as in any other neural network. The whole network architecture up to this point is shown in figure 1 .

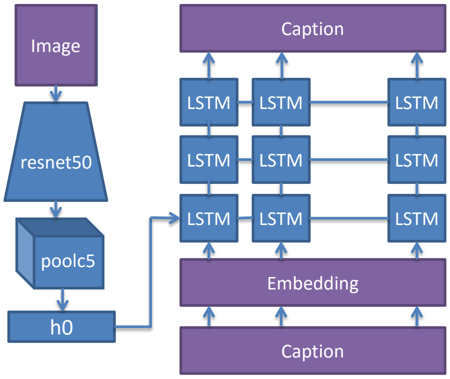


Figure 1. Captioning network without attention

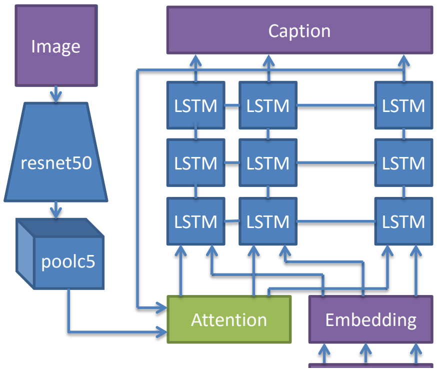


Figure 2. Captioning network with attention

3.3. Attention models

In the previous section, we assumed that the spatial dimensions of the CNN image features were averaged together. Now, we describe a method to weight these spatial locations according to their perceived importance. This is known as an "attention mechanism." We accomplished this by adding an "attention gate" to the LSTM architecture. The new "attention LSTM" equations are

$$\begin{pmatrix} i \\ f \\ o \\ g \\ a_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \\ \text{softmax} \end{pmatrix} \circ W \begin{pmatrix} I \circ a_{t-1} \\ h_{t-1} \\ x \end{pmatrix}$$

where a_t represents the attention parameters at time t. We can see that the attention parameters define a linear transformation of the input image. Furthermore, the softmax function ensures that for all elements we have $a_t > 0$ and for the whole vector we have $\|a_t\|_1 = 1$. In other words, the attention weights are positive and sum to one. The new architecture with attention is shown in figure 2.

4. Experiments

In this section, we evaluate the proposed network architectures on a real image captioning dataset.

4.1. Dataset

There are many datasets for image captioning. We originally planned to caption videos with the newly-released MVAD dataset [6]. However, due to computational and algorithmic limitations, we decided to limit the scope of our project to still images. We decided to use the Common Objects in Context (COCO) dataset from Microsoft, which is among the most widely-used for this task [4]. The dataset consists of about half a million images, split into training, validation, and test sets, along with human annotations for the training and validation sets. Each annotation is a sentence of around 10-20 words, and there are 5-7 annotations per image.

4.2. Metrics

Given the huge volume of our data, cannot possibly show the results for each image. Thus, we need a way to quantify the average accuracy of the system on the whole dataset. There are several metrics by which to judge the quality of machine-produced text, and none without criticism. We chose to use the Bilingual Evaluation Understudy (BLEU) metric, as it is one of the simplest and best known. Before describing the BLEU score, we will describe a simpler and better known metric, called the "precision." Let x be a vector of machine-produced n-grams, and let y be a vector of ground truth n-grams. For example, x could be the words in a sentence describing a movie frame, with x_i representing an individual word, and y could be words from scripts describing the same scene. We often wish to have a ground truth y representing the many possible linguistic translations of the same idea. The precision is

$$p = \frac{1}{N} \sum_{i=1}^{N} 1\{x_i \in y\}.$$

The BLEU score is similar to the precision, except that each occurrence of an n-gram in y can account for only one occurrence in x. For example, the statement "The the the the the the" would receive a perfect precision if the reference translation contained the word "the," but not necessarily a perfect BLEU score, as we are limited to counting only as many occurrences of "the" as appear in y.

4.3. Results

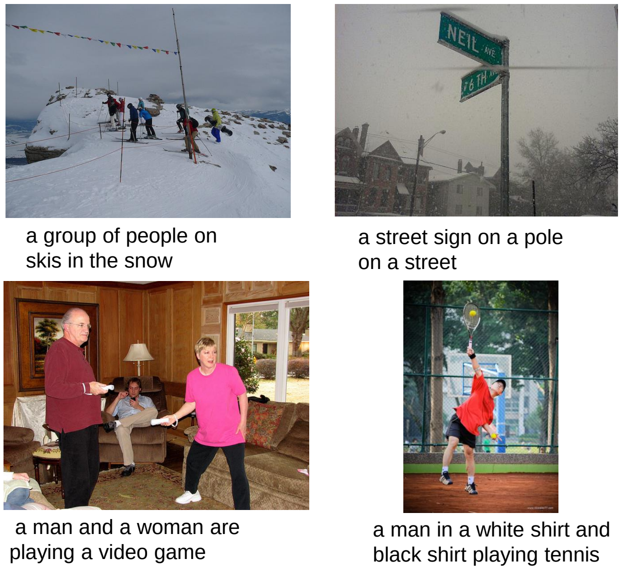
The results in our experiments so far are encouraging. After trying our different combinations of hyperparameters, we trained the basic LSTM model from figure 1 with two layers of LSTMs, a hidden state of length 2048, and a word embedding of length 1536. The length of the generated sequence was 51 words, the maximum length of all sequences in the COCO dataset. Image features of length 2048 were extracted using a pretrained ResNet-50 model [1]. To optimize the LSTM model parameters, we used the Adam descent method in batches of 64 pairs of images and captions [3].

To make the classification problem feasible, the vocabulary was restricted to the 2,000 most common words from the COCO dataset. All other words were replaced with a special "unknown" token. All tokens less than 51 words in length were padded with a special "null" token, signifying that no more words are needed to describe the image. Note that the 51 word limit includes "start" and "end" tokens enclosing each caption.

After these modifications, the model was trained for 10,000 batch iterations, or approximately 1.5 epochs, on the full COCO training dataset of approximately 400,000 images. We used a pretrained Caffe model for the ResNet CNN, and implemented the LSTM model in Torch.

The qualitative results of this model are quite interesting. For many images, the model generates informative and grammatical captions. Figure 3 shows some of the most impressive results. While many captions are informative, some describe a scene completely different than the one in the image, while others are complete gibberish. Although we cannot rigorously separate the two tasks of the network, we might call the former type of error a failure in image recognition, and the latter a failure in text generation. It should be noted that most gibberish captions make liberal use of the "unknown" token. Figure 4 shows some examples of these kinds of failures.

To qualitatively assess the accuracy of our model, we report the world-level BLEU score, as discussed in section 4.2. All start, end, null tokens are stripped from both the predicted and ground truth captions prior to computing BLEU. Note that we include the unknown tokens when computing BLEU, which is strongly to our disadvantage. A graph of the BLEU score over time is shown in figure 5. We can see that there is not much improvement beyond the first few hundred iterations. It could be that the model has derived all possible benefit from our fast learning rate, and is unable to descend steep valleys. Or, it could be that the model bounces back and forth between local minima,



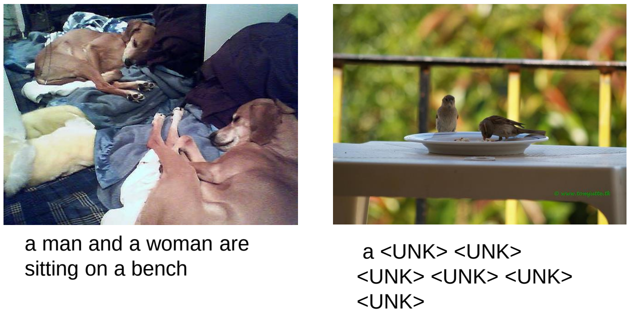
a group of people on skis in the snow

a street sign on a pole on a street

a man and a woman are playing a video game

a man in a white shirt and black shirt playing tennis

Figure 3. Successful captions generated by our neural network



a man and a woman are sitting on a bench

a <UNK> <UNK> <UNK> <UNK> <UNK> <UNK>

Figure 4. Failure cases from our neural network.

all of which are nearly as good. The good news is that the beak BLEU score of 0.572 at iteration 4000 is fairly high. This can be interpreted as saying that 57% of the generated words correctly describe the input images.

Finally, we show an example output of the attention network architecture from figure 2. The paramters were the same as before, except we used only one layer for simplicity. At this time we do not have a full evaluation of the performance differences between that and the prior LSTM architecture. However, we can see in figure 6 that the caption is informative, and the attention on the word “baseball” seems to be in the part of the image with the baseball player and glove.

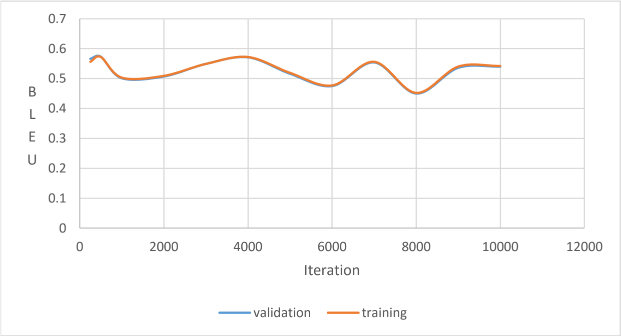


Figure 5. Word-level BLEU score for the network without the attention model. The training and validation accuracies are nearly identical.

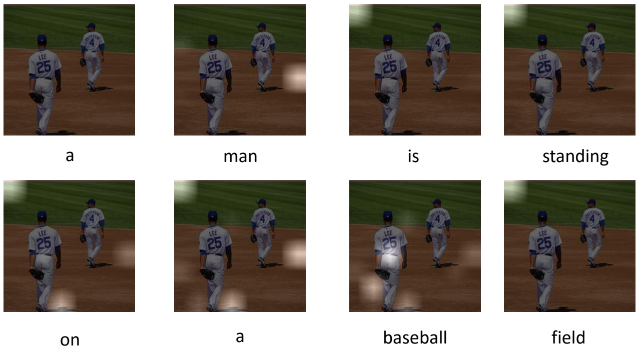


Figure 6. Attention captioning example. White areas have higher attention weights.

5. Conclusion

We have developed a neural network architecture for image captioning and reported its accuracy on the COCO dataset. We have also extended the architecture to use attention models, and showed some initial results from this investigation. The current experiments are encouraging, giving a reasonably good BLEU score with little training.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[2] A. Karpathy and F. Li. Deep visual-semantic alignments for generating image descriptions. *CoRR*, abs/1412.2306, 2014.

[3] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[4] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context.

In *Computer Vision–ECCV 2014*, pages 740–755. Springer, 2014.

[5] A. Rohrbach, M. Rohrbach, W. Qiu, A. Friedrich, M. Pinkal, and B. Schiele. Coherent multi-sentence video description with variable level of detail. In *Pattern Recognition*, pages 184–195. Springer, 2014.

[6] A. Rohrbach, M. Rohrbach, N. Tandon, and B. Schiele. A dataset for movie description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3202–3212, 2015.

[7] A. Torabi, C. Pal, H. Larochelle, and A. Courville. Using descriptive video services to create a large data source for video annotation research. *arXiv preprint arXiv:1503.01070*, 2015.

[8] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko. Translating videos to natural language using deep recurrent neural networks. *arXiv preprint arXiv:1412.4729*, 2014.

[9] K. Xu, J. Ba, R. Kiros, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. *arXiv preprint arXiv:1502.03044*, 2015.

[10] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. *arXiv preprint arXiv:1511.06984*, 2015.