LSTM for Image Annotation with Relative Visual Importance

Geng Yan[1]

Yang Wang[2]

Zicheng Liao[1]

[1] College of Computer Science
Zhejiang University

[2] Department of Computer Science
University of Manitoba

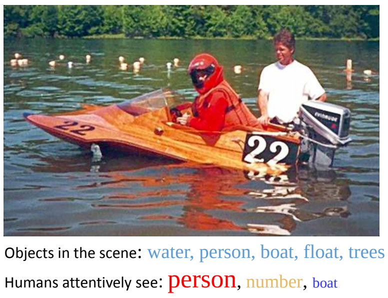
Objects in the scene: water, person, boat, float, trees
Humans attentively see: person, number, boat

Figure 1: An image may contain a rich set of objects, e.g. person, boat, tree, water, helmet, etc. But when humans are asked to describe an image, they do not enumerate all the objects in the image. Instead, they will choose a few important objects and put them in the some order depending on the relative importance of these objects. In this paper, we develop a method for generating such ranked list of object tags that take into account of the relative object importance.

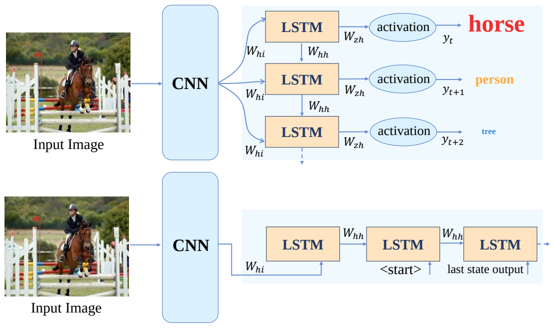

Figure 2: Illustration of the LSTM model. (Top) In our model, the image feature is used as an input to the LSTM at each time step. (Bottom) In the LSTM model used for image captioning, the image feature is only used to start the initial state in the LSTM model.

We consider the problem of image annotations that takes into account of the relative visual importance of tags. Humans have the remarkable ability to selectively process very narrow regions of the scene that are important to us. So when asked to annotate the image in Fig. 1, we only mention a subset of the objects appearing in the image, and we mention the important objects first. In this paper, we propose a method for producing such ranked tag list for a given image. Such a ranked tag list can be useful for various applications including image retrieval, image parsing and image caption generation.

Our proposed approach combines the convolutional neural network (CNN) for images and the LSTM for sequential data. First, we extract a feature vector from the given image using CNN. In this paper, we use a convolutional neural network to represent the image, and use a LSTM model to represent the ordered tag list. Our model is inspired by some recent work on using RNN models for image captioning [1]. But the difference is that in image captioning, the image feature only directly modulates the start state of the RNN used for generating the captions. Once the first word in the caption is generated, the remaining words in the caption are generated purely based on the hidden states of the RNN and the image feature is no longer used. For sentence generation, this is appropriate since the words in a sentence tend to have very strong dependencies. It is reasonable to use the image features to only start the initial state of the RNN and let the RNN model take care of generating each word in a sentence. However, we found this RNN model is insufficient for our application. The reason is that words in a sentence tend to have very strong dependencies, so it is possible to predict the next word based on previous words in a sentence. In contrast, although the tags in a tag list have some loose dependencies, they are not strong enough for us to predict the next tag in the list purely based on previous tags. To address issue, we modify the RNN model so that the memory cell at each time step also takes the image feature as its one of the inputs. Fig. 2 illustrates our model and compare it with the RNN model for image captioning.

Image representation: Following prior work (e.g. [1]), we represent an image as a 4096-dimensional CNN feature vector using pre-trained VGGNet. We then use a fully connected layer to reduce the dimension to d. In other words, given an input image I_m, we represent it as a d-dimensional feature vector as:

$$I = W_I \cdot CNN(I_m) + b_I \tag{1}$$

where $W_I \in \mathbb{R}^{d\times 4096}$ and $b_I \in \mathbb{R}^d$ are the parameters to be learned. $CNN(I_m)$ is the 4096-dimensional CNN feature extracted on the image I.

LSTM for tag list prediction: We modify the standard LSTM, so that the hidden state at each time step considers the image feature $v(I)$ as one of the inputs. In other words, our LSTM model is defined as follows:

$$i_t = \sigma(W^{(i)}I + U^{(i)}\mathbf{h}_{t-1}) \tag{2}$$
$$f_t = \sigma(W^{(f)}I + U^{(f)}\mathbf{h}_{t-1}) \tag{3}$$
$$o_t = \sigma(W^{(o)}I + U^{(o)}\mathbf{h}_{t-1}) \tag{4}$$
$$\tilde{\mathbf{c}}_t = \tanh(W^{(c)}I + U^{(c)}\mathbf{h}_{t-1}) \tag{5}$$
$$\mathbf{c}_t = f_t \odot \mathbf{c}_{t-1} + i_t \circ \tilde{\mathbf{c}}_t \tag{6}$$
$$\mathbf{h}_t = o_t \odot \tanh(\mathbf{c}_t) \tag{7}$$

At each time step t, we need to predict a tag from a vocabulary of size V. We use another linear layer to project the hidden state h_t into a vector of dimension V, followed by a softmax operator. This will give us the probability of choosing each of the V possible tags as the predicted tag at time t:

$$\mathbf{z}_t = W^{(z)}\mathbf{h}_{(t)} + \mathbf{b}^{(z)} \tag{8}$$
$$p_{t,v} = \frac{\exp(z_{t,v})}{\sum_{k=1}^{V}\exp(z_{t,k})} \tag{9}$$

where $\mathbf{z}_t \in \mathbb{R}^V$, and $p_{t,v}$ denotes the probability of picking the v-th tag in the vocabulary as the predicted tag at time t.

Model learning: Let I_m be an image in the training set, and $\mathbf{y} = [y_1, y_2, .., y_T]^\top$ denotes the corresponding order tag list of length T. We define the following loss function on this training instance:

$$\ell(Im, \mathbf{y}) = \sum_{t=1}^{T}\sum_{v=1}^{V} \mathbb{1}(y_t = v)\cdot \log(p_{t,v}) \tag{10}$$

The loss on the whole training set is simply the summation of the loss on each training instance. The parameters of the model can be learned by minimizing the loss function using stochastic gradient descent. We do not explicitly use a regularization term in Eq. 10.

We demonstrate the effectiveness on the PASCAL2007 dataset and the LabelMe dataset.

[1] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.