International Institute of Information Technology
Bangalore

Visual Recognition

AI 825

Part 2: Mini Project

Group Members: **Team Id 7**
Mukul Gupta (IMT2020083)
Ujjwal Agarwal (IMT2020128)
Deep Shashank Pani (IMT2020129)
Amar Pratap Singh (IMT2020131)

May 15, 2023

1 Part A

Problem: Design a CNN-LSTM system (preferably in pytroch) that can perform image captioning based on the following details: and LSTM architecture that best suits your case.

Image captioning is a process where a network is trained on a set of images along with some texts describing the scene in the image. Typically a CNN- RNN network is used to encode the image features and predict the captions in a recurrent manner through the RNN. In this report we present an Inception-LSTM based network that extracts the latent features from the image using a pretrained inception network. It is followed by encoding the latent features and sequence features to generate a concatenated encoding which is finally passed onto the decoder to predict the next token in the sequence. Hyperparameter tunings have been reported along with some example images from the test images and their captions predicted by the trained model.

About the dataset: Flickr8k dataset is a widely used benchmark dataset in the field of computer vision and natural language processing. The dataset comprises approximately 8,000 high-quality images obtained from the photo-sharing website Flickr. Each image in the dataset is associated with five human-generated captions, resulting in a total of 40,000 captions.The images in the dataset are diverse and depict a variety of scenes and situations.

Pre-processing: First, the captions underwent a transformation into lowercase, accompanied by the elimination of punctuation marks. To facilitate the retrieval of captions based on image names, a dictionary was established. In this dictionary, the image names served as keys, while the corresponding lists of five captions acted as their corresponding values. To prepare the captions for training, a <start>token was added at the beginning, and an <end>token was appended at the end of each caption belonging to the training images.

1.1 Model Architecture

The model consists of three parts -

1. **Image Encoder**
   We used Inception v3 to extract latent features from an image.The reason of choosing Inception is because its ability to leverage multi-dimensional filters, allowing it to learn features across various scales. By doing so, it acquires a comprehensive understanding of the image's content, enabling the generation of descriptive captions that effectively represent the scene portrayed in the image. This attribute of Inception facilitates the extraction of rich representations, contributing to the generation of meaningful image descriptions.

2. **Sequence Encoder**
   The sequence encoder receives an input sequence, which is padded to a maximum length of 38. Within this sequence, the <pad>token's ID is included based on the size of the actual sequence. Subsequently, the sequence includes the <start>token's ID, followed by the token IDs representing the words in the sequence. Lastly, the <end>token's ID marks the sequence's conclusion. This entire sequence is then forwarded to the embedding layer, responsible for generating an embedding that captures the semantic essence of each token.

3. **Decoder**
   In order to generate the next token in the sequence, the decoder fulfills the task of utilizing the encoded representations of both the image and the sequence. It accomplishes this by amalgamating the encoded vectors of the image and sequence, resulting in a concatenated representation with a dimension of 2 times the hidden size. Subsequently, the concatenated encoding undergoes a transformation through a linear layer with 'relu' activation, which reduces its dimension to the hidden size of 256. Finally, the transformed encoding is passed through a softmax layer, producing a vector with a dimension equivalent to the vocabulary size. This vector represents the probability distribution of the subsequent token within the input sequence.

1.2 Sequence Prediction

Beam search was used to predict the sequence prediction instead of greedy search as it might happen that one token would give a high probability currently but the overall probability in the subsequent token predictions might go down. And there could be another token that has low probability in the probability distribution currently, but the subsequent token predictions achieve a higher combined probability.

The beam search algorithm starts with continuously evaluating the collection of the k most favorable sentences at each iteration, serving as potential candidates for generating sentences of size t + 1. Subsequently, it retains only the top k sentences among the outcomes, as this approach offers a closer approximation to the likelihood of achieving the ultimate highest value.

For **evaluating** the quality of captions, BLEU (Bilingual Evaluation Understudy) score was used that gave a score in range from 0 to 1.The score serves as a measure of similarity between the given text and the reference text. Values closer to 1 signify a higher degree of similarity between the texts. Achieving a perfect score in practice is challenging since a translation would need to precisely match the reference text.
s **Trained image embeddings: Link**
Part B: Modified baseline
We implemented the attention as a modification for both vision and word embedding for better scores. Since the original method is called Attention Word Embedding, we are going to explain it with that name.

2 Attention Word Embedding (Part B)

Attention Word Embedding(AWE) consists of two components. First, AWE uses a variant of self-attention to narrow down relevant words in the context fzor the prediction of the masked word. In contrast to prior work,the attention weights are a function of both the context words and the target/masked word. Second, AWE embeds the context words and the masked word representations in a shared subspace, since both representations embody the meaning of the word. Note that this is in sharp contrast with the baseline's embedding methods which use Common Bag of Words(CBOW), which learns two different embeddings for each word, one when the word is present in the context and another when it is masked. AWE augments the CBOW model with the attention mechanism in two different ways. First, we introduce two new matrices, a key matrix,

$K \in R^{N \times D^{'}}$ and a query matrix, $Q \in^{N \times D^{'}}$, where N is the vocabulary size and $D^{'} \in Z$. With the attention vector, the context vector $\mathbf{c}$ is not modeled as a simple sum, but rather as a weighted sum of context word vector embeddings.

$$c_{attn} = \Sigma_{i \in [-b,b]-\{0\}} a_{w_i} u_{w_i}, \tag{1}$$

where a_{w_i} is the attention weight of each context word vector u_{w_i}, calculated using the key matrix K and the query matrix Q.

$$a_{w_i} = exp(k_{w_0}^T q_{w_i}), \tag{2}$$

Second, we share weights between the context word embedding matrix and the mask embedding matrix in our model, we set $U = V$. In CBOW, U and V are seperate matrices for better performance. However, in AWE, the intuitive choice to have U and V same works better and adds interpretability to the model. On top of that, AWE has lesser parameters than CBOW, because the key and query matrices are much smaller than CBOW's value matrix $V = [u_1, u_2, ..., u_D]^T \in R^{N \times D^{'}}$.

3 Objective Benchmarks

The models have been benchmarked on BLEU4 and METEOR scores.

	BLEU4	METEOR
Baseline	0.49	0.436
Modified baseline	0.564	0.503

4 Sample Outputs

Baseline

A man wearing a hat.

****

A white cat sitting.
****

Modified baseline

A man wearing an orange hat.
****

A man sleeping on a bed.
****

5 Subjective Comparisons

The trained image captioning model made predictions that accurately provided the high level overview of the various moving parts in the picture. The model achieved a bleu score of 0.48 and higher in most of the captioning. It accurately described the scene in almost all of the images. The model only shows some limitations, when it sees some words occurring more often in a given context than others during training. The LSTM inside the network learns the high correlation between these words. And so when the model is provided a context which has an object that had high correlation with some other object in the training data (like 'dog' and 'run'), the model is more likely to make biased predictions even when the actual context in the image corresponds to some other term (like 'dog' and 'fight'). However the model was able to perform well overall, keeping in mind that the number of training samples was just 6000 images. Inception network learned the context in the image in detail by leveraging filters of various dimensions. Using GloVe embeddings proved to be useful as the model did not have to learn embeddings from scratch which reduced the number of epochs during training for convergence. Ultimately, the model was able to predict captions with a high bleu score of 0.48 and higher in most of the images.

Attention improves the scores in several ways. Studies have shown that the attention mechanism helps a neural network to attend to relevant features in the input (Wiegreffe and Pinter, 2019; Vashishth et al., 2019). Thus, motivating the integration of the attention mechanism in CBOW. We visually analyze the attention weights to investigate if the attention mechanism models the importance of a context word for the masked word prediction in AWE. The investigation revealed two key findings. First, no matter the masked word, the attention weight of the masked word and a highly frequent word is typically high. Even though the attention weight is large, the similarity between word vectors is very close to zero, thus not affecting the probability of prediction of the masked word.Second, if we leave out these highly frequent words from the set of context words, we observe that the attention weights focus on more informative words in the context for the prediction of the masked word. However, if we need to focus on highly frequent words with high correlation we end up having similar edge cases. Our bleu scores has now been increased to 0.54 and higher.