hw4 visions

Justin Hughes

April 2020

1 RNN application

The first application that comes to mind is how it can be used in game implementation (I was thinking of chess). RNN is suitable for this problem simply because RNN works well with seuence prediction and chess moves are represented as sequences with more or less correct responses to certain moves and positions much like text prediction. We can use RNN as move prediction and training an agent to make strong moves based on the current position of the board. The board could be represented as images or as a non-visual model (list, dictionary, tree or tensor). When making a chess agent we could consider using this with the minimax/alphabeta algorithm along with previously played which optimizes the move options based on what the algorithm decides using minimax/alphabeta with guidance of the previously played games (using RNN). How I imagine the algorithm might work is assigning weights to moves from a specific position and as the minimax progresses, it uses moves as individual cells in the network which I believe would be a one to many application. A possible result of one move prediction might be 'The Queen is threatening to take the knight with a discover check by moving the bischop.' as a move sequence.

2 Attention

Attention is used to emphasize on the important features of an image by finding relevance between the image and the captions and then using the more relevant pieces to create an appropriate response. This algorithm was proposed as a solution to noisy images which could be problematic for RNN, especially with large sentences. One idea that might help this algorithm is some sort of reinforcement between convolutional feature extraction to help with initializing the weights for the RNN with attention step. Another idea might be stratisfying sentences to help the algorithm break down captions and possibly have a better idea to connect outputs to specific areas of the captioning.

3 Evaluation Metrics

BLEU Metric:

The BLEU metric is essentially a weighted average over the candidate sentences and the references. It begins by describing the connection between candidates and references are for the most part stronger when there are frequent matches use of vocabulary. So the program begins by finding the number of matches between the n-grams of the candidate and the n-grams of the references. It goes on to discuss the precautions that the algorithm requires such as modifying the n-grams, length of the sentences, sentences that contain similar items but have vague meaning (problem with recall), etc. The issue with BLEU is potentially starting with poor references and decreasing the result of the algorithm. Another potential issue could be having poor initialization with the references. For example if all referenced sentences are shorter than all candidate sentences and the candidates are penalized.

CIDEr Metric:

First the we take a set of candidate sentences and all a set of reference sentences that are human responses to the image (in the experimentation they use 50). The sentences are converted into n-grams which are fully reduced ordered words. In the paper they limit the size of the n-grams to 4. Then compare the candidate n-grams to the reference n-grams in triplet annotation. That is compare two candidate n-gram to a reference sentence and determine which one is most similar to the reference sentence. We do this by calculating Term Frequency and Inverse Document Frequency which is used with the cosine similarity to determine the CIDEr score. This is when they reduce the number of candidates by giving lower weights to the ones that don't appear in the references and ones that are too vague in the references (occur to all images). Since the algorithm is dependent on human responses, we are comparing the accuracy of the algorithm to the accuracy of how humans describe the image which leaves room for error. If there is a lot of noise in the data or is unclear by some means, there might be mixed results and throw off the accuracy of the algorithm.