CS224N Assignment 4:
Reading Comprehension on the Stanford Question Answering Dataset

Shashwat Udit
School of Engineering
Stanford University
Stanford,CA 94305

March 22, 2017

Abstract

In this assignment, a deep learning architecture was created for use on a reading comprehension task. The specific task was finding the answers to queries in a paragraph of contextual information, with the dataset used being the Stanford Question Answering Dataset. The centerpiece of the architecture were bidirectionnal lstms that were used to encode the question and context representations, which were combined into a context vector, which when weighted was used to predict the beginning and end of the answer spans. Owing to limitations in time and personnel, many means remain to potential improve the predictave power of the model, which obtained an F1 score of 5.165 on the dev set, and these potential improvements are described in the paper below.

1 Introduction

1.1 Dataset

The Stanford Question Answering Dataset(SQuAD) has 100k triplets consisting of questions, answers, and a context paragraph in which the answer may be found. The context paragraphs come largely from Wikipedia, and the questions and answers were extracted by humans, and human accuracy at the task remains unsurpassed although recent models have shown impressive results[1234] at this and other reading comprehension challenges.

1.2 Implications for the Model

Unliked named entity recognition which requires sorting into a few predefined categories, question answering is much more open-ended, with any of the words in the context paragraph possibly being the beginning of the answer and the answer can simply be the beginning word or stretch to the end of the context paragraph. Determing the answer span therefore requires the model to learn the complex interplay between words and ideas in the query and in the context paragraph in their entireity, and keep in perspective important features as it searches the context paragraph. This need, to remember and prioritize is a natural fit for LSTMs, and so they were the main components of the model implemented.

2 The Current Model: The Sequence Attention Mix Model

The first decision made was how to take in the data. Initially a batch size of 10 question context paragraph sets was used, which was shruken down to 5 when it was suspected that memory issues might be behind the failure of a submission that was successful on the development set on the test set, though that was ultimately unsucessful. In each batch, the questions and context paragraphs were padded with zeros to the maximum length of a question or paragraph in the batch respectively. The now padded questions and paragraphs were then fed into to be ecoded, which was done by LSTMs. These LSTMS were bidirectionnal in order to take into account both the forward and backward context of each part of the questions and paragraphs. After being encoded by the LSTMs, the respective outputs where then combined to create the context vector.

To deode our reprsentation and extract the answer span, the context vector was then combined with the paragraph representation. Then using trained weights, the paragraph reprsentation concatentated with the context vector was classified as to the probablity of being the start point of the answer span, and using a diffirent set of weights the various points were estimated as to the likelihood of them being the ending point of the answer span. The training of these weights and other parameters was done using an Adam optimizer, with loss being computed via softmax versus one-hot vectors that consisted of the actual start and stop of the answer spans withing a paragraph.

3 Potential Upgrades to the Model

There were a number of potentail upgrades that were not implemented to due lack of additional time or personal required to deal with debugging, segmentation faults, memory issues or other complications associated with more sophisticated models before the project deadline, though not necessarily for lack of effort. The first and foremost priority would be to add another bidirectional lstm to the model, this time in the decoder. Usings weights has limitations because of their inabiligy to grasp connections and context throught the paragraph. In fact, occasionally they would predict the end of the answer span at point previous to where weights had predicted the beginning. Although code was written to reverse them in those eventualities, they illustrated the need for a decoding mechainism that can take into account the whole context paragraph.

There's also no reason do stop at adding just one mechanism to the decoding. Althogether alternative models could be added as well, with a vote being taken of the various members of the ensemble to make the prediciton of the answer spam. Other reaserachers have gotten increased accuracy even with highly effective models by using an ensemble.

In addition to adding complexities of the architecture, there are simpler ways to boost peformance. One of the very simpliest would be using additional training time to go over the entire training set. Care would be needed to prevent overfitting to the training data, so likely dropout would need to be addded to the training.

Another option would be to take more time tune the learning rate, hyperparamters, and to explore other optimizers than the Adam Optimizer. In spite of many other obstacles, exploding graddients did not noticebally occur in the span of the lifetime of this model. However, if they do occur after some altherations, they could be taken into account with gradient clipping.

Additionally, if memory issues were not a problem, additoinal input information could be used. In the bidirectional attention flow paper, reseraches included not just word embeddings but charcter embeddings for every charcter. Less drastically, in the current model the default GLoVe embedding was used, 100.npz, but a larger size could be used to improve performance if run-time and memory issues could be resolved.

Acknowledgments

This project was made possible thanks to genorosity of Microsoft which donated time on Azure GPUs for training, and the faculiy and course staff of CS224n, without whose assistance it would not have possible.

References

[1] Zhigou Wang & Wael Hamza & Raidu Florian. (2017) Bilateral multi-perspective matching for natural langugage sentences.*arXiv preprint arXiv:1702.03814, 2017.*

[2] Danqi Chen. & Jason Bolton & Christopher Manning. (2016) A thorough examination of the cnn/daily mail reading comprehension task.*arXiv preprint arXiv:1606.02858.*

[3] Caiming Xiong. & Victor Zhong & Richard Socher (2016) Dynamic coattention networks for question answering. *arXiv preprint arXiv:1611.01604.*

[4]Shuohang Wang. & Jing Jiang (2016) Machine comprehension using match-lstm and answer pointer.*arXiv preprint arXiv:1608.07905*