Text Based Information Retrieval (B-KUL-H02C8B) Visual Question Answering Assignment 2018-2019

Graham Spinks, Guillem Collell and Marie-Francine Moens

This assignment is only relevant for students of the 6 study points group. The first part of the assignment is due March 27, 2019, the second part May 23, 2019.
Graded for 1/3 of points.

The assignment may be conducted in teams of 1 to 3 people.

1 Introduction

The mission of Visual Question Answering (VQA) is to answer open-ended questions about images. This task is interesting as there is a wide spectrum of possible applications for visual understanding, ranging from robotics to information retrieval and AI assistants. It is also challenging however as it requires a mechanism to combine relevant information from different types of data. A very accurate system is still quite far off as it requires common-sense general knowledge as well as deep semantic understanding and reasoning with both the visual and written input. However, with recent advances in deep learning some progress has been made to achieve some results in VQA.

An important decision in any machine learning application is the chosen representations for the data. Text, for example, can be represented as dense real-valued vectors, derived using various training methods inspired from neural-network language modeling. Images on the other hand, are often deconstructed into a vector of features by using Convolution Neural Networks (CNN).

In this project you will tackle the task of VQA in two steps: in part I, you will build an informative representation of text in a multi-task setting: question answering and autoencoding. In part II, you will enhance your question answering system by adding visual features to your network.

2 Programming Assignment Description

Source code (in C++, Java, MatLab, Python, ...) must be submitted in Toledo by the dates specified above.

Important:

- Add commentary.
- Add description of how to run your software.
- Add your test examples.
- If asked in the assignment, add documentation with the actual results and comparisons of different models.

2.1 Part I. Representations for text

Question Answering can be interpreted as the task of generating a prediction in a sequence: Given a sequence of words (that make up the question), which are the most probable words that follow (that make up the answer to the question)? While several approaches are possible to generate sequences, recurrent networks, and more specifically Long Short Term Memory networks (LSTMs), are the most common solutions in recent years. At each step, they input and process the next sample (word, character, ...) but also maintain information about what they have previously perceived. The LSTM learns what information might be useful for its task and conserves it in its hidden state. Thus, the value of the hidden state of the LSTM after feeding an input sentence (e.g. a question) to it, can be interpreted as a representation of the sentence. The ability to perform a particular task (e.g. answer a question) on the obtained representation, determines the quality or value of the representation.

In the context of an autoencoder, the goal is to build a compact representation that can be used to reconstruct the input. The network that creates the representation is then referred to as the encoder, while the network that reconstructs the input is called the decoder. In the context of an LSTM that learns to autoencode text, the compact representation is the value of the hidden state after feeding the input sentence through it. In order to train the autoencoder, the LSTM is encouraged to reconstruct the input sentence on the basis of the learned representation as good as possible.

In part I your focus will only be on text. Head to the following location:

https://www.mpi-inf.mpg.de/departments/computer-vision-and-multimodal-computing/research/vision-and-language/visual-turing-challenge/

where you can download the 'Question answer pairs - train' and 'Question answer pairs - test' files for the DAQUAR dataset. You will implement a neural network of your choice that creates a representation of the text with the aim to perform 2 tasks at once:

1. Reconstruct the question from the representation
2. Generate a suitable answer for each question

As the network gets better at reconstructing the question in subtask 1, it means that it is learning a representation that contains relevant information about the input sentence. Since the questions generally require visual data for an accurate answer for subtask 2, there is a limit to the accuracy of the answers for this part. The network should be able to guess the correct answer in some cases however. Note that your solution should build one representation that is suitable for both subtasks, i.e. at least the network that encodes the input sentence should be shared for both subtasks. Again, we have used the example of an LSTM above, but you may choose any network you want.

For each subtask you will need to define an objective function to optimize. A loss that generally works well when predicting sequences is the cross-entropy loss. Assume you have a vocabulary of 5 words (w1, w2, w3, w4, w5). The output of your network predicts the probabilities for the next word in the sequence. For example, if the probabilities are p = (0.5,0.1,0.2,0.1,0.1), w1 is the most likely next word. However while training, you actually know what the next word is in the ground truth. Assume the next word in the ground truth is w3, then q = (0.,0.,1.,0.,0.). By comparing p and q with the cross-entropy objective function, the network can improve the prediction p to approach the distribution in the ground truth during training. Consult your particular neural network framework to find out how to feed the data to the objective function.

After training, evaluate the performance of your network on both subtasks. For the autoencoder calculate the accuracy of the output. For the question answering, evaluate the accuracy and also the WUPS score with threshold $t = 0.9$ as formulated in [1]. Note that the WUPS score requires an ontology, so use the WordNet ontology (https://wordnet.princeton.edu/). The easiest way to import WordNet is with the nltk.corpus package if you choose to use python (https://www.nltk.org/). You may refer to this implementation of the WUPS score.

Your algorithms will be judged and graded on the basis of originality, elegance, efficiency (scalability), functionality and performance. Given that you have the gold labels of the test set, please refrain from fine tuning in the test set (we will not give you any extra points for that!). On the day of the demo, you will be provided a set of new questions to test your implementation on.

Part I is due on March 27, 2019. Please provide your code and a short report (max 1 page text + additional figures are allowed).

2.2 PART II. Adding visual features

This part will be communicated in March.

3 Grading

Your grade for the programming assignment will consist of: 50% for the text-only implementation (PART 1), and 50% for the text + image implementation (PART 2).

References

[1] M. Malinowski and M. Fritz, “A multi-world approach to question answering about real-world scenes based on uncertain input,” in *Advances in neural information processing systems*, pp. 1682–1690, 2014.

[2] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[3] M. Malinowski, M. Rohrbach, and M. Fritz, “Ask your neurons: A neural-based approach to answering questions about images,” in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1–9, IEEE Computer Society, 2015.