Homework 3: Visual Question Answering

Introduction

Our task was to solve a VQA problem, generating answers to questions about images. Therefore we joined two models: a CNN for extracting a features vector from a given image and a LSTM model to deal with the sequence of questions focusing through an attention mechanism on the key aspects of the vector of features.

Network Design

We tested two different approaches for the CNN part. The first approach involved the usage of a VGG model, finetuning the last 6 layers and keeping frozen the first 54 layers. As the computational effort required to finetune more layers was significant and due the fact that the majority of the weights were freezed, our model was not able to perform at best. Therefore, we implemented a light hand-crafted model based on a series of convolutional blocks with 3x3 filters and ReLU as activation followed by a MaxPooling with 2x2 filters. We then concatenated the output of the CNN and the LSTM. Finally we selected the answer, as in a standard classification problem, performing a softmax over all the possible outcomes.

Text Embedding

First of all, for every single question, we prepended <sos> and removed the special characters ‘?’ and ‘ ‘. Then we used the Tokenizer provided by Keras library to encode the questions. Finally we padded them to the maximal question length. On the other hand, we created a dictionary for the answers, treating them as classes assigning an integer to each one.

Data Preparation

In order to handle all the images, avoiding memory issues, we created a CustomDataset. This generates a dictionary of pairs image-encoded_question and images are resized to 256x256x3 inside the class. Then we defined the batch size and we spilt the dataset into train and validation. We tweaked the value of the batch size from a grid and selected 32 as final choice.

Optimization parameters

As we tackled the problem like a classification task, we used a categorical crossentropy as loss function. We used Adam as optimizer starting from a learning rate 1e-3 for the first epochs and manually decreasing it to 1e-4 for the last 3 epochs.

Conclusions

We obtained a final classification score over the test set equal to 0.617. It could have been interesting, with more time and computational power, to implement another approach based on a CNN pretrained model, finetuning a higher number of layers.