POLITECNICO DI MILANO

MSc in Computer Science and Engineering

Artificial Neural Networks and Deep Learning

A.Y. 2020/2021

CHALLENGE 3
VISUAL QUESTION ANSWERING

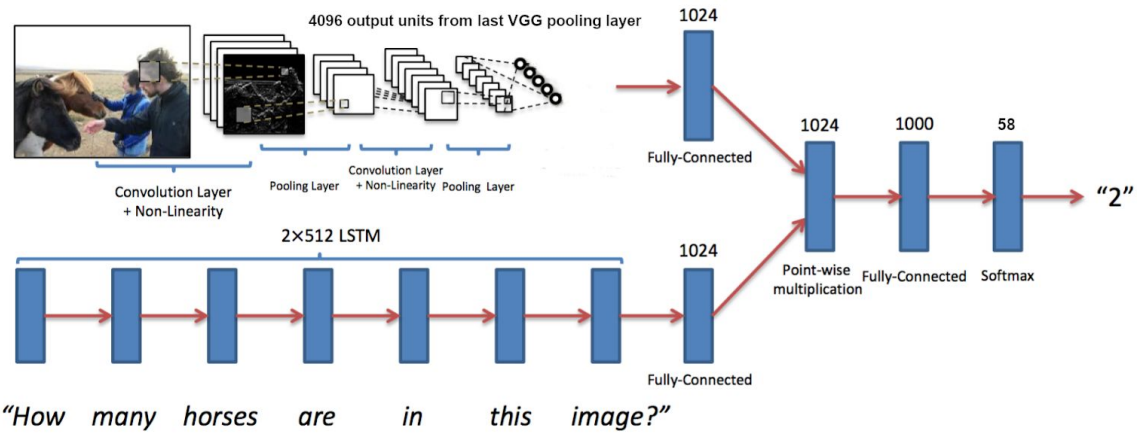
Team: *Spatial Extent*

Nicola Rosetti 940435

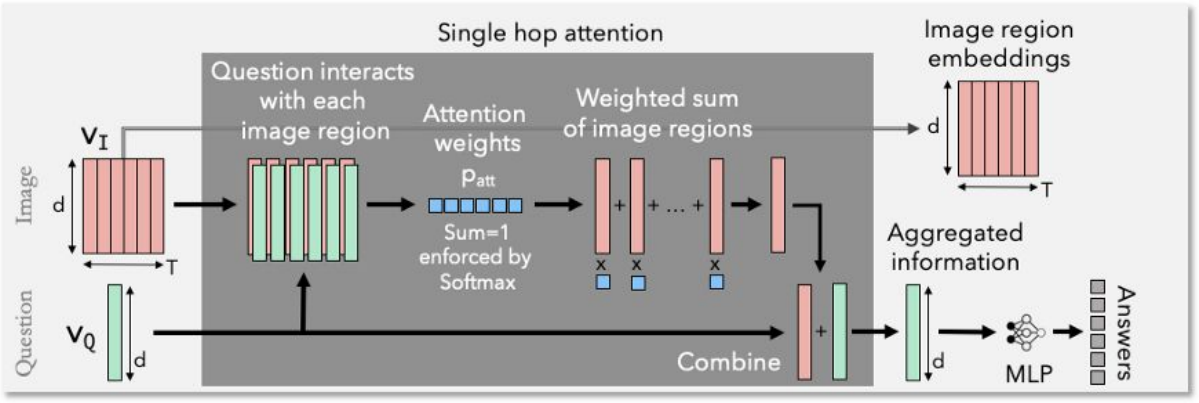
Alessio Russo 945781

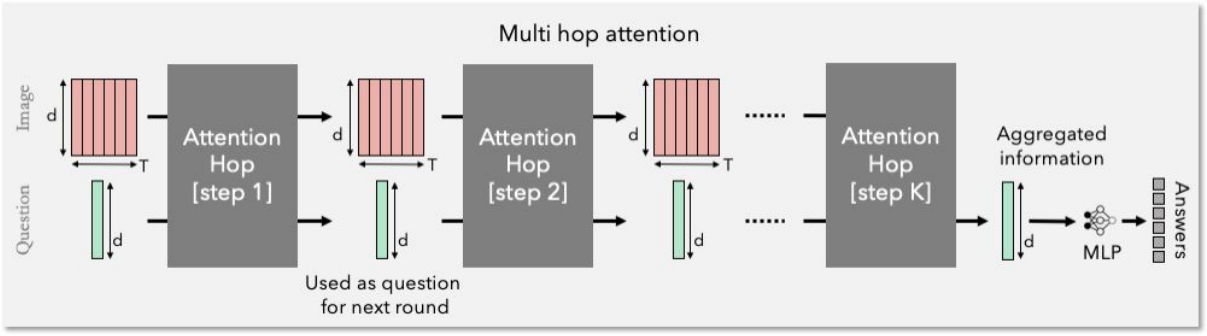
Vittorio Torri 945208

For this challenge we started from a model composed of two parts: one pre-trained convolutional (VGG19) and a recurrent one with an embedding layer (learnt from scratch) and 2 LTSMs layers. After these two separated blocks, their outputs (the CNN feature map and the LSTM final state) are point-wise multiplied in order to determine the answer, found by solving a classification task over 58 different possible answers (FFNN). The loss function used is the sparse categorical crossentropy with Adam optimizer.



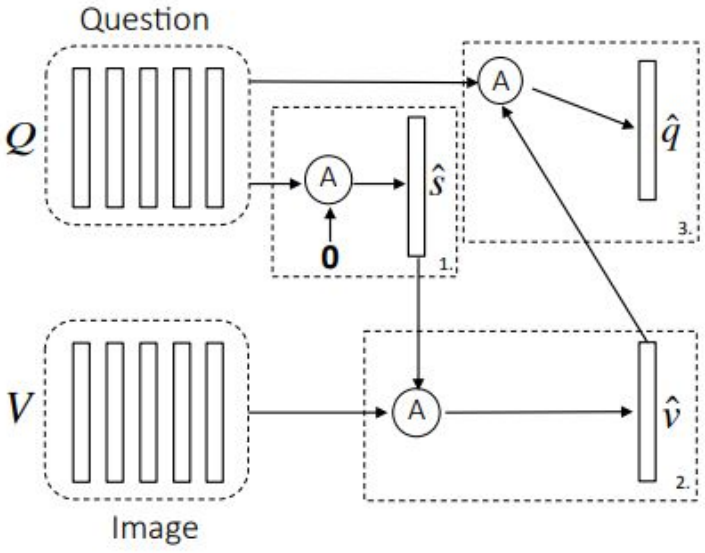
This model was quite good from the beginning, achieving more than 0.6 accuracy on both validation and kaggle test, with a batch size of 32 and lr=10^(-3).
We tried to use Glove instead of the learnt embedding and we obtained a small improvement.
Here we resized the images to 224x224, but we observed only a small difference.
We tried to change the number of layers of the recurrent part (1,2 or 3), the units in those layers (256-512) or the number of dense layers in the final classification network, in order to reduce the number of parameters and therefore trying to reduce overfitting. We also tried different dropout rates. Performance increased a bit up to 0.6235 on validation set.
In order to really try to perform better, we started using attention mechanisms, with two methods: stacked attention and co-attention.
The first one uses the question features to determine attention over the image and then combine the question with the attentioned image. This is done multiple times in a stack:





We tried again different combinations of structures, with 1-2-3 stacked attentions, different sizes and numbers of LSTMs, with and without Glove, changing dropout rates between 0.2 and 0.5. We also tried to use a stack of LSTMs or GRUs with residual connections and residual connections around attention layers. The best here was achieved using a single LSTM with 512 units, embedding with size=300, 2 stacked attention layers (size=512), dropout=0.4, bs=32, lr=5*10^(-4). For the best result we also applied a small data augmentation, with small shifts, small zoom and horizontal flip. We know that this can change some answers, but, exploring the dataset, we considered minimal the amount of them and effectively it gave a small benefit here.This achieved 0.6466 on validation set.

The second attention model is the co-attention model. We initially implemented it considering only word attention (in the alternated version), not the entire hierarchical attention proposed in the paper.



We tried variations similar to the one tested in the previous models, we also tried to consider image features coming from lower levels of VGG, to use ResNet50v2 as backbone and to add Batch Normalization after the CNN, but none of this was beneficial. We tried to finetune a different number of layers, hoping that the CNN would adapt to drawings instead of real pictures, but performance worsened since a lot more parameters needed to be trained.
We considered cutting the questions to 15 words, since few of them had more, but the impact on the parameters was negligible.

Considering the large unbalancing among the answers in the dataset we tried to use a focal loss with various smaller weights for ‘yes’ and ‘no’ answers (the most common), but it didn’t help. We also tried RMS-Prop instead of Adam.
We considered different schemes for dropout, arriving to remove it inside the co-attention layers, where instead L2 was beneficial.
The best result was achieved with a single 256 LSTM, attention layers with size=256, dropout=0.5, L2=10^(-7), bs=32, lr=4e-4, Adam, training for 23 epochs, no data augmentation, leading to 0.6591 on validation set.
We also implemented the full hierarchical version, but the model was too complex and therefore it overfitted too much.
At the end we verified that our model was not predicting only ‘yes’ and ‘no’ and maybe the numbers (most common answers), but it seems to be able to consider all the answers, even if a more balanced training set would certainly improve the result. The final retraining has been done on 99% of the dataset, for 29 epochs leading to an accuracy of 0.67372 on Kaggle.