Artificial Neural Networks and Deep Leaning

Homework 3: Visual Question Answering

Giovanni Dispoto, Matteo Sacco

January 31, 2021

We started creating a VQA Model as described in the Paper VQA: Visual Question Answering

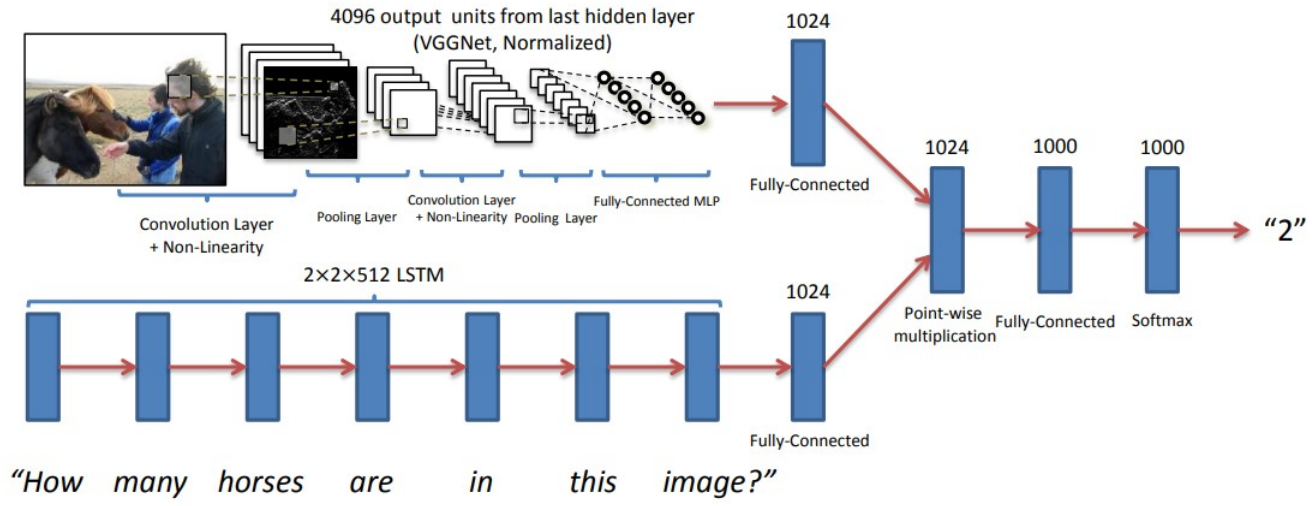

Figure 1: *VQA Model from the Paper*

We first tried to keep VGG16 in the CNN but after experimenting with others pre-trained feature extraction models we chose Xception.

We adjusted the *CustomDataset* class in order to return the information as: img, question_tokenized, answer in one hot encoding. Then we started training the model obtaining low accuracy on validation set (≈ 0.48) before it started ovefitting. We tried also to tune hyperparametrs using a Grid Search approach with HParams, but without success.
After some days we have discovered that this was due to the fact that the dataset was using only 8 bit integer for questions, whiched messed with the word's token.

Having fixed this issued we now tried to add data augmentation by using *Albumentations*. After some research online and some *trial and error* by training the model and observing the performances, we settled with the data augmentations functions reported below. At this point we obtained a validation accuracy of ≈ 0.6

```python
import albumentations as A
A_transform = A.Compose([
  A.HorizontalFlip(p=0.5),
  A.Blur(p=0.2),
  A.Downscale(scale_min=0.5,scale_max=0.9,p=0.15),
  A.GaussNoise(p=0.3),
  A.ElasticTransform(p=0.2, alpha=120, sigma=120 * 0.1, alpha_affine=120 * 0.1),
  A.Rotate(p=0.4, border_mode=0, limit=40),
])
```

Listing 1: Albumentation functions

By observing the dataset we noticed that the answers was very unbalanced toward yes/no answers.

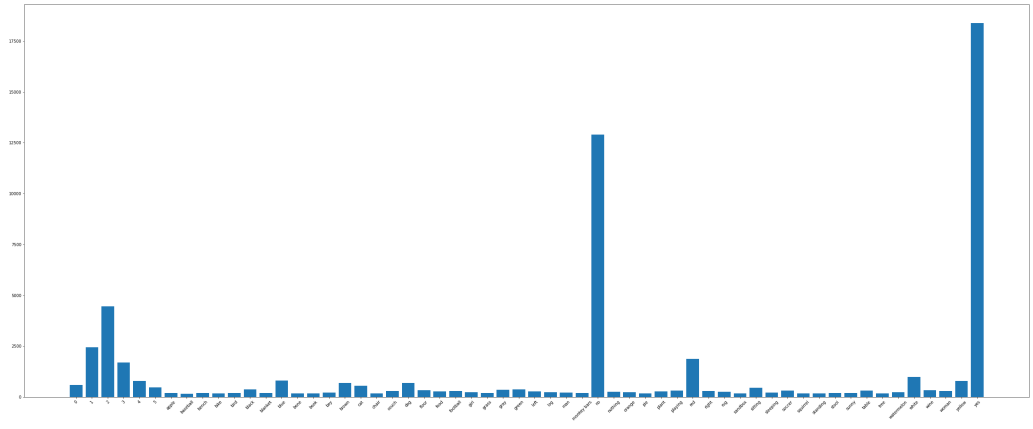

Figure 2: *Answer Frequencies*

Thefore, we tried to modify class weight using the following formula

$$w_i = \log(\mu \cdot \frac{total\_samples}{samples_i})$$

and passing them to the fit function with $\mu = 5$ obtaining a notable improvement reaching $\approx$ 0.63 on the test set.

We think that the model proposed can be improved by tuning the hyper-parameters for instance by using a Grid Search, but training session were long and we unfortunately ran out of time.