IRSE Project assignment 1

Maarten Sijmkens (r0616538)

April 2, 2021

1 Network architecture

For the projection model I took inspiration from [1]. Both networks are a sequential model of these layers: Linear/BatchNorm/ReLU/Dropout/Linear, followed by L2 normalization. Their implementation can be found in `models.py`. The dropout has a probability of 0.5.

After some experiments I used a latent space dimension of 32 and a hidden dimension of 2048. This was a tradeoff between memory usage and performance of the model.

2 Loss function

With the given loss function in the assigment, using multiple hyperparameter settings, the loss did decrease. However the accuracy on the validation set would not increase at all. See figure 1. Thats why I created a modified version of the loss function:

$$l_{IT} = a_1 \mathbb{1}[s_{ij} = 1] |f(x_i) - g(z_j)|_2^2 + a_2 max(0, ((1 - s_{ij})c - |f(x_i) - g(z_j)|_2)^2)$$

This function has the objective to bring the distance between an image and a caption pair with similarity 1 to 0. Pairs with a similarity less than 1 will have the objective to have at least a certain margin between them. This margin scales by how unsimilar the pair is. Both this modified loss function as the loss function given in the assigment are implemented in `losses.py`.

3 Hyperparameters

I found that a batch size of 256 works best with a learning rate of 5e-5, decayed every 10 epochs with 10%. For the margin c and tradeoff parameter $\alpha = 1 - a_1 = a_2$ I did multiple experiments where both parameters took a value of 0.25, 0.5 or 0.75. After these experiments, the MRR on the validation set was best when $\alpha = 0.5$ and $c = 0.5$. The implementation of MRR can be found in `metrics.py`.

4 Results

With the modified loss function, $\alpha = 0.5$ and $c = 0.5$, we obtain a MRR on the test set of 0.013. This is not ideal but certainly not terrible in a set of more than 27000 images. The learning history of this model can be found in figure 2. A similar result is obtained by using the same parameters, but using intermodal distances by setting $\beta_2 = \beta_3 = 0.25$.

5 Running

To start training run `train.py`. Optional parameters can be found in the beginning of the file. The script will store all relevant information in a new folder in the `exp` directory. After training you can check the MRR on the test set by running `test.py`. Make sure to edit the experiment name in the beginning of the file to your training experiment's folder. This script will also store all projection, needed for retrieval. Run `retrieve.py` with the query as parameters to retrieve. Again edit the experiment name in the beginning of the file and make sure you have run `test.py` on this experiment.

References

[1] Lazebnik, Liwei Wang Yin Li Svetlana. *Learning Deep Structure-Preserving Image-Text Embeddings*

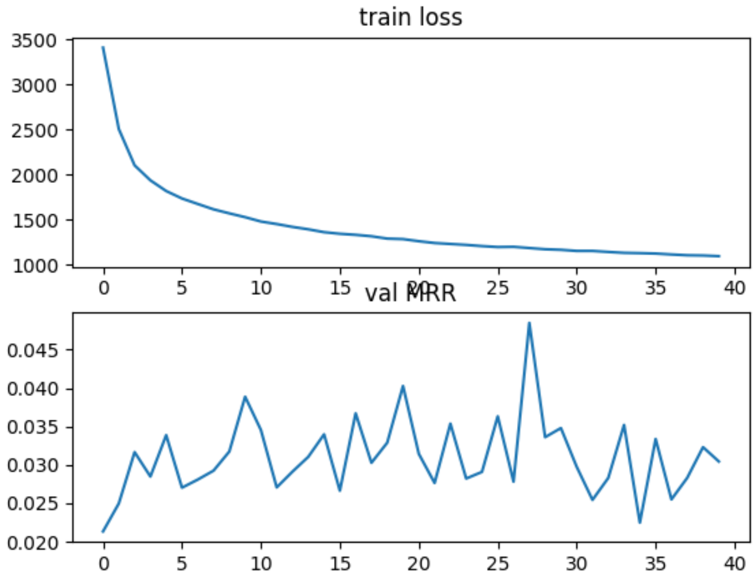

Figure 1: Training results with assigment loss and $\alpha = 0.5$ and $c = 0.5$

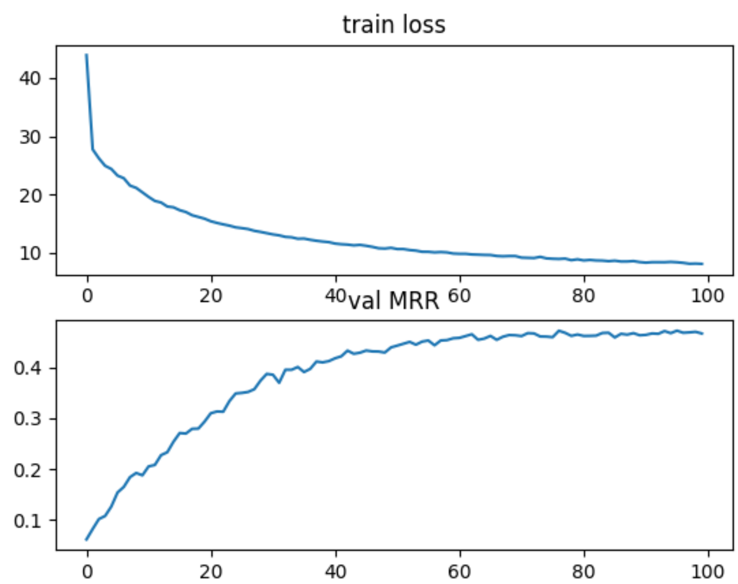

Figure 2: Training results with modified loss and $\alpha = 0.5$ and $c = 0.5$

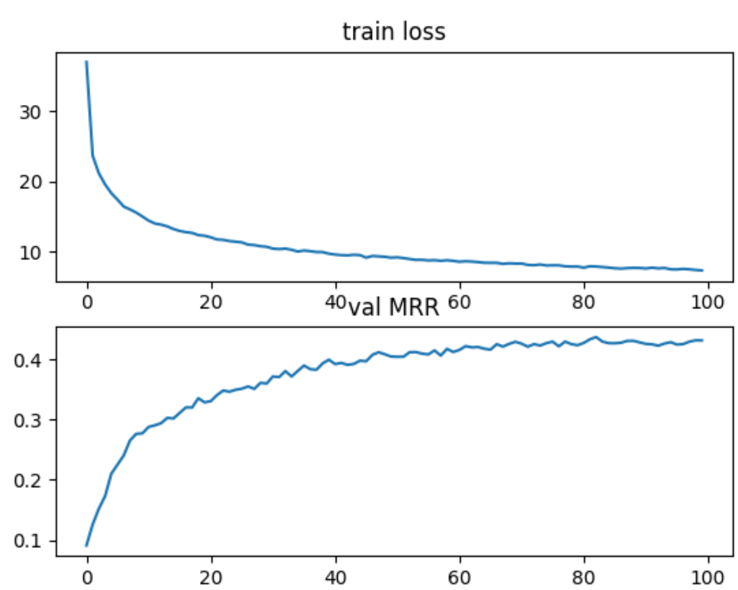

Figure 3: Training results with modified loss and $\alpha = 0.5$, $c = 0.5$ and $\beta = 0.5$