CS698O: Mid Term Report
Modified Stacked Attention Networks for VQA using Visual grounding of Phrases

Abhishek Verma(14026) , Ayush Tulsyan(14167)

November 25, 2017

1 Pre-planned objective

We had planned to implement the paper[1] in PyTorch by the time of mid-term review. The only codebase available for this paper was implemented in Torch and Theano.

2 Targets achieved

We have successfully implemented the paper [1] in Pytorch. The model works well and gives results close to those described in the original paper. Github repository

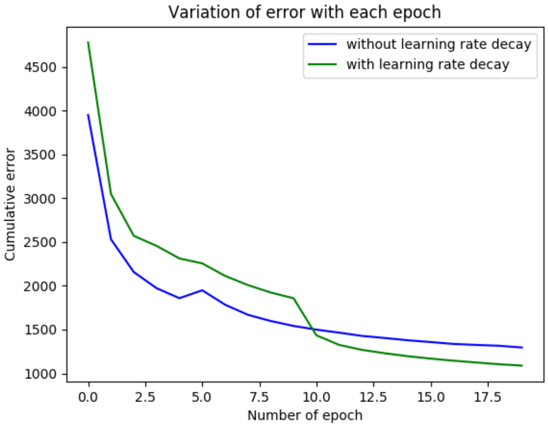
3 Implementation details

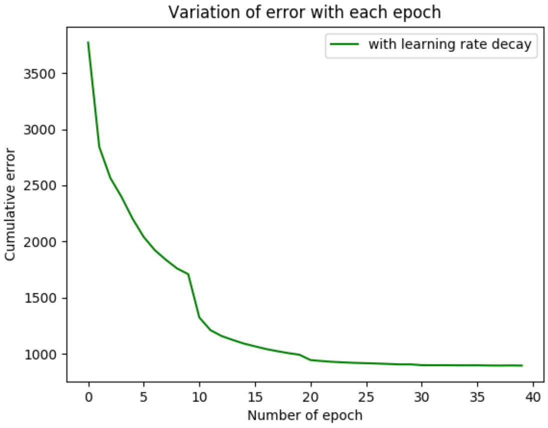
We first went through this Torch code which is implemented in torch and tried to understand the code. Once we had a proper understanding of it, we proceeded with our implementation. Implementation details are as follow:

> qDownloaded the MS-COCO dataset from visualqa website and pre-trained VGG model from here. Filtered the train dataset and removed the questions whose answers didn't lie within the top 1000 most frequent answers and removed corresponding images too. Extracted 196 feature vectors of 512 size from every image by passing each image through the VGG model. Represented each image as a feature vector of size 196 x 1024 by passing the feature vectors through a linear layer. Converted each question phrase to a feature vector of length 1024 using LSTM(tried both single as well as double layers). This was done to ensure that question and visual embeddings are in the same semantic space. Passed image representation vector and question representation vector through stacked attention network to generate a 1000 size probability array(number of distinct answers for the questions were 1000). Trained our model using different set of hyper-parameters(we are still optimizing them).

4 Plots

Plots for training error of models using different hyper-parameters are as follow:





5 Results

We tested our model on MS-COCO validation dataset for visual question answering. Without using diminishing learning rate, accuracy was around 32% while with diminishing learning rate in place, accuracy increased to 43%.

6 Inferences

- Training error decreases and accuracy increases with an increase in number of lSTM layers till a point after which it starts increasing.
- Training error decreases when number of attention layers were increased from one to two. However it started increasing as the number of attention layers were increased further from two. The case with accuracy is vice versa.
- Training error started increasing after 25th epoch when we kept the learning rate constant. We analyzed it and started diminishing the learning rate at as the epochs proceeded. This resulted in an almost constant training error after 30th iteration.

7 Further plan of action

- Replacing the LSTM model for question embedding by CNN model.
- Using pre-trained LSTM model(trained on Googles billion words dataset) for question embedding model.
- Using attention in LSTM.
- Introduction of a semantic embedding module(two-layer-perceptron with Dropout inserted in between the layers), i.e. output of visual encoding would be made to pass through it
- We continue to read about visual grounding and question answering as much as we can. If we find any other appropriate model, we will try to implement it too.

References

[1] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng and Alex Smola. *Stacked Attention Networks for Image Question Answering.* arXiv:1511.02274