Report: Visual question answering system

Model Architecture:

- CNN Architecture: The given image features were sparse, hence used CNN to reduce size and get dense feature maps
    - Maxpooling to reduce image size from 14*14 to 6*6
    - 1*1 Conv to reduce number of feature maps from 512 to 128
- Divide the resultant image features:
    - The 6*^*128 image features are divided into 4 blocks of 3*3*128, This is done to preserve the depth of the image and use local features of image individually to make predictions
    - Each block is combined with summarized question (output of encoder) and fed to an LSTM
    - LSTM ensures that local features are treated as a sequence. Hence combining the results of each local block to get compact representation of the question and image
- Make predictions using softmax for each answer word
    - This compact representation (hidden state of above LSTM) is again fed into another LSTM to predict answers
    - Final layer conatins a softmax, that predicts a one hot encoded answer for each answer word.
    - LSTM with Softmax was chosen over a multi-label multi-class classifier. This is because class_weights were easier to calculate and gave better results than multiclass.
- Class_weights:
    - Since each answer word is considered as a class, the data is highly imbalanced. Wors like “chair” appear more frequently in answers
    - To address this, sklearns compute class weights feature has been used to compute weights of each answer word.
- Output Vocab:
    - Output vocab was restricted to only answers in training set
    - Since each answer is considered a different class. The model needs positive and negative examples of all classes. Since this was the case only for words in training answers, it was decided to limit output vocab to only these
    - This also greatly reduced the number of parameters required for final model

Important Features:

- This model takes advantage of local features along the dept. Hence for a given question each block of image is examined independently by the model and the results are combined to get a global representation of the image
- The model uses ~5M trainable parameters.
- A normal neural network would have (~2 M) parameters (+ones from autoencoder and CNN) considering same image compressions. It would also lack local and global connectivity.

Improvements:

- Model parameters can be further reduce if autoencoder is removed
- However, this would require more training
- Text data was linearly separable. Can make a simpler model for final prediction
- Increase output vocab using online training: all vocab (maybe from wiki) and predict=>add to training data=> train strategy

Rough sketch of mode architecture (Appendix)