Assignment 4

Aman Bhardwaj (2019ISY7580)

Anjali (2019CSZ8763)

1. Non-Competitive Part

(a) Encoder:

We designed a CNN based encoder that handles the variable sized images. We implemented ResNet50 CNN architecture. We did Andrew Ng's cnn course from coursera and applied concepts learned from there to implement RestNet50.

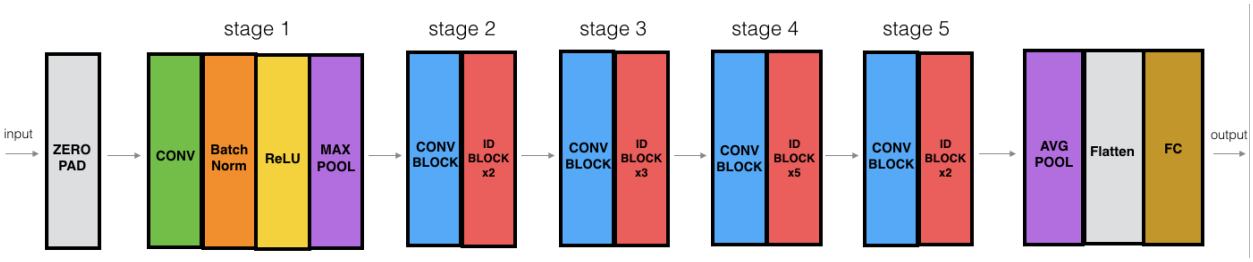

Figure 1 : Resnet50 architecture

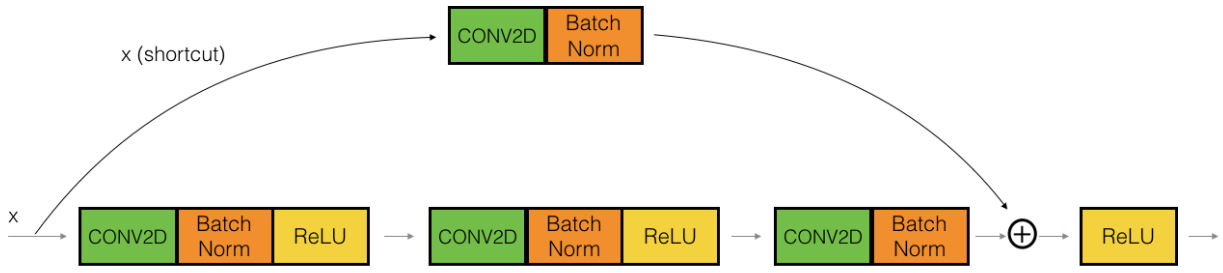

Figure 2 : skipping connection helps handling washing gradient

Here we take advantage of skip connection because while propagating the gradient from last layer to first layer, vanishing gradient issue occurs and initial

layes learn less. So these skip connections help in handling gradient vanishing so that initial layers also learn properly[1].

(b) Decoder:

We implemented an LSTM based decoder which takes output of encoder as hidden state and <start> token as initial word and predicts the next word and again uses this predicted word as input for next word generation and so on. This way this decoder generates the captions given the encoded image input. We used word emabading to convert each word in emabading form and then used in LSTM.

(c) Training the model:

We trained our model on google colab using cuda environment from scratch.

(d) Testing:

We generated captions using greedy (softmax) as well as K beam search approach and results are submitted in respective files.

2. Competitive Part

We used a pretrained Resnet50 model here for the competitive part.

References

[1] https://towardsdatascience.com/understanding-and-coding-a-resnet-in-keras-446d7ff84d33

[2] http://static.googleusercontent.com/media/research.google.com/en//pubs/archive/43274.pdf

[3] https://machinelearningmastery.com/teacher-forcing-for-recurrent-neural-networks/

[4] https://arrow.tudublin.ie/cgi/viewcontent.cgi?article=1013&context=airccon

[5] https://web.stanford.edu/class/cs224n/readings/cs224n-2019-notes06-NMT_seq2seq_attention.pdf