A Neural Image Caption Generator

Presentation By: Ankit Kumar, Avinash, Daksh Saraf, Rachneet Kaur
Table of Contents

- Implementation
- Parameter Exploration
- Results
  - Qualitative Results
  - BLEU scores
Implementation

● Training and Testing dataset used: MSCOCO 2014 dataset
  ○ ~80K Training images
  ○ ~40K Testing images
  ○ 5 captions per image

● Loss function:
  ○ Cross entropy loss

● Optimizer:
  ○ ADAM (initial learning rate = 0.01)
  ○ SGD with momentum 0.9 (initial learning rate = 0.1/0.01)
  ○ Adaptive learning rate decay

● Early stopping
Architecture Details

- **Encoder:**
  - Pretrained (on ImageNet) Resnet 50, Resnet 101
  - Embedding dimension: 512
  - Training only last 2 layers of Resnet (linear and batchnorm)

- **Decoder:**
  - 1, 3, 5 and 10 layers LSTM and GRU
  - No. of hidden units experimented in RNN = 512 and 1024
  - Maximum sequence length = 25

- **Batch Size = 32**
Implementation

- Scoring function:
  - BLEU 1
  - BLEU 4

- Data Augmentation
  - Resize: 224 X 224
  - Horizontal Flip
  - Vertical Flip
  - Normalization

- Inference Sampling
  - Greedy search
  - Beam search (Beam sizes tried = 5)

- Vocabulary Threshold: 5

- Epochs trained:
  - ~25 training hours for each model variant with ~2.5 hours for each epoch
  - ~3 testing hours for each model

- Computational Hours used:
  - Blue Waters: 720 GPU hours
  - Google Colab: 40 GPU hours

- Regularization:
  - Dropout
## Variants Tried

<table>
<thead>
<tr>
<th>Encoder</th>
<th>Decoder</th>
<th>LR</th>
<th>Decoder Layers</th>
<th>Optimizer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resnet 101</td>
<td>GRU</td>
<td>0.1 with decay</td>
<td>1</td>
<td>SGD with momentum/ADAM</td>
</tr>
<tr>
<td>Resnet 50</td>
<td>LSTM</td>
<td>0.01 without decay</td>
<td>3</td>
<td>SGD with momentum</td>
</tr>
<tr>
<td>Resnet 50</td>
<td>LSTM</td>
<td>0.1 with decay</td>
<td>1</td>
<td>SGD with momentum</td>
</tr>
<tr>
<td>Resnet 50</td>
<td>LSTM</td>
<td>0.1 with decay</td>
<td>3</td>
<td>SGD with momentum</td>
</tr>
<tr>
<td>Resnet 50</td>
<td>LSTM/GRU</td>
<td>0.1 with decay</td>
<td>5</td>
<td>SGD with momentum</td>
</tr>
<tr>
<td>Resnet 50</td>
<td>LSTM/GRU</td>
<td>0.01 with decay</td>
<td>7</td>
<td>SGD with momentum</td>
</tr>
<tr>
<td>Resnet 50</td>
<td>LSTM/GRU</td>
<td>0.01 with decay</td>
<td>10</td>
<td>SGD with momentum</td>
</tr>
</tbody>
</table>
GRU vs LSTM for 1 Layer
Early Stopping

Model 1:
Resnet50
RNN unit: LSTM
1 layer
512 Hidden units
SGD with momentum
Training time: ~25 hours
Testing time: ~3 hours

Trained for 25 epochs, best performance at 22 epochs.
The BLEU score achieved with 1 Layer LSTM by using Resnet 50 as encoder on MS COCO dataset is 20.34 without beam search.
LSTM with 10 Layers

The loss seems to be pretty high compared to the single layer LSTM decoder and hence we did not proceed further with this model.
GRU Decoder with 5 Layers

1. The loss after 20 epochs was 1.95
2. The BLEU score using Greedy Search was 21.40 and using Beam Search was 21.94
3. This is the current best model we have.

** The results are for beam width of 5 - exploring other beam widths is one of our tasks further
# Results of other architectures

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Loss</th>
<th>BLEU</th>
<th>BLEU (Beam)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM - 5L</td>
<td>2.17</td>
<td>20.41</td>
<td>21.41</td>
</tr>
<tr>
<td>GRU - 7L</td>
<td>2.09</td>
<td>21.41</td>
<td>20.33</td>
</tr>
<tr>
<td>LSTM - 7L</td>
<td>2.96</td>
<td>18.58</td>
<td>20.54</td>
</tr>
</tbody>
</table>
Predicted: a black and white picture of a man on a horse drawn carriage

Predicted: a street sign with a picture of a car and a street sign

Predicted: a fire hydrant on a sidewalk next to a street.
A person sitting on a wall ledge near the water with a bike next to him and two other people sitting together on the wall a short distance from him.

A large clock hanging off the side of a building.

A wooden shelf on a brick wall with a storage bin under it, a laptop and lamp are on the shelf.
Future Tasks

1. Replicate the study on other datasets: Flickr8k/Flickr30k
2. Transfer learning from the model trained on MSCOCO to PASCAL
3. Calculate METEOR and CIDER scores for our implemented models
4. Try different beam sizes (3, 4, 6, 7) for Beam Search in Inference
5. Distributed training on BlueWaters
6. Try more regularization techniques (L2 weight decay for ADAM)