
Image Captioning using an LSTM network

Fangzhou Ai

Department of Electrical and Computer Engineering
University of California, San Diego
La Jolla, CA 92093
faai@eng.ucsd.edu

Yue Qiao

Department of Electrical and Computer Engineering
University of California, San Diego
La Jolla, CA 92093
yuq021@eng.ucsd.edu

Zunming Zhang

Department of Electrical and Computer Engineering
University of California, San Diego
La Jolla, CA 92093
zuz008@eng.ucsd.edu

Abstract

In this report, we ...

1 Introduction

Automatically generating a natural language description of an image, a problem known as image captioning, has recently received a lot of attention in Computer Vision. The problem is interesting not only because it has important practical applications, such as helping visually impaired people see, but also because it is regarded as a grand challenge for image understanding which is a core problem in Computer Vision. Generating a meaningful natural language description of an image requires a level of image understanding that goes well beyond image classification and object detection. The problem is also interesting in that it connects Computer Vision with Natural Language Processing which are two major fields in Artificial Intelligence.

In this assignment, we will explore the power of Recurrent Neural Networks to deal with data that has temporal structure. Specifically, we will generate captions for images. In order to achieve this, we will need an encoder-decoder architecture for this assignment. Simply put, the encoder will take the image as input and encode it into a vector of feature values. This will then be passed through a linear layer for providing the input to the LSTM. It will be trained to predict the next word at each step. We used a pre-trained convolutional network as the encoder and an LSTM model as the decoder and tried to fine tune the encoder and train the decoder by backpropagating error into it. The error will come from caption generation. The training uses images and several captions for each image generated by humans. We then run the network in generative mode to generate captions on images it has never seen before. We also tried to replace the LSTM in the encoder part with Vanilla RNN and GRU and compared the performances of them.

2 Related Work

3 Methods

3.1 Dataset

For this image captioning task, we used the dataset from the well-known Common Objects in Context (COCO) repository. COCO is a large-scale object detection, segmentation, and captioning dataset. In this report, we used a subset (around 1/5) of the COCO 2015 Image Captioning Task. The training set contains around 82k images with roughly 410k captions while the test set has around 3k images with almost 15k captions. The original images in the dataset are of different sizes and aspect ratios, which we are resizing to 256x256 before the training.

3.2 Model

3.2.1 Baseline Model

Our captioning system is implemented based on a Long Short-Time Memory (LSTM) network (baseline model). For the encoder part, we use a frozen pretrained convolutional network, namely ResNet50, as the encoder. We removed the last layer of pre-trained and added a trainable linear layer with outputs a feature vector of a fixed size for each image. This set the initial state of the LSTM network based on the image. For baseline model, we resized the image to 256x256 and hidden size of 512.

3.2.2 Vanilla RNN and GRU

For the model comparison, we also tried Vanilla RNN and GRU. For these two models, we simply replaced the LSTM module in the encoder with either a Vanilla or a GRU, while the others remained to be same. Then, trained and compared the performance of these three different models.

4 Results

4.1 Learning Curve

The learning curve for the Vanilla RNN and GRU could be found in Figure 1.

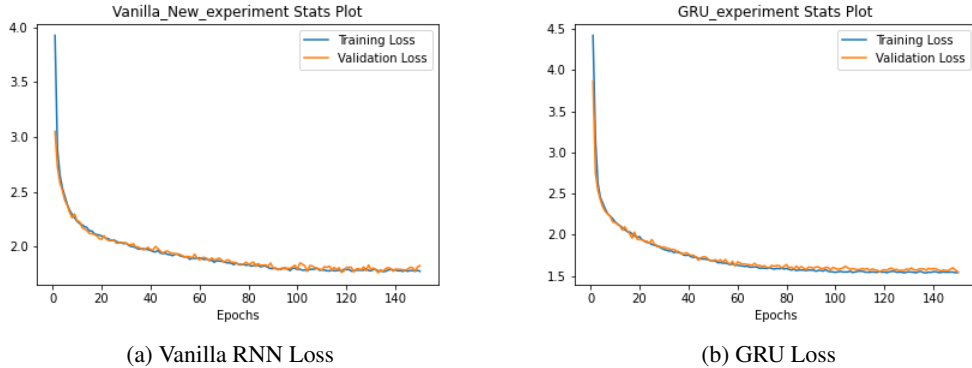


Figure 1: Training and Validation loss for Vanilla RNN and GRU

4.2 Cross Entropy Loss

The test set Cross Entropy Loss of different could be found in Table 1

Table 1: Cross Entropy Loss for different models

Model	Testset Cross Entropy Loss
Baseline	TBD
Vanilla RNN	1.819
GRU	1.586
Fine tuning model	TBD

4.3 BLEU Score

For this report, we set the weight for BLEU1 to be [1, 0, 0, 0] and BLEU4 to be [0.25, 0.25, 0.25, 0.25]. The BLEU scores for different models could be found in Table 2

Table 2: BLEU scores for different models

Model	BLEU1	BLEU4
Baseline	TBD	TBD
Vanilla RNN	54.59	8.93
GRU	65.16	17.17
Fine tuning model	TBD	TBD

4.4 Images Visualization of the Best Performance Model

5 Discussion

5.1 Baseline with different Temperatures

5.2 Baseline LSTM vs Vanilla RNN vs GRU

5.2.1 Learning Curve

5.2.2 Cross Entropy Loss

5.2.3 BLEU Score

5.3 Baseline model vs fine tuning model

5.3.1 Learning Curve

5.3.2 Cross Entropy Loss

5.3.3 BLEU Score

6 Individual contributions to the project

Fangzhou Ai

Yue Qiao

Zunming Zhang