CIS581: Computer Vision and Computational Photography
Final Project
Final Project Report

Tianlin Du (dtl1995)
Wentao He (wentaoh)
Yezheng Li (yezheng)

December 17, 2018

1 Introduction

In this (short) project, we will implement vanilla recurrent neural networks (RNNs) and Long-Short Term Memory (LSTM) RNNs and apply them to image captioning on COCO.

Note: this project is adapted from the Standford CS231n course.

2 Goal

We will take a look at an interesting multi-modal topic where we will combine both image and text processing to build a useful Deep Learning application, aka Image Captioning. Image Captioning refers to the process of generating textual description from an image based on the objects and actions in the image.

Here are some examples of image captioning:

Figure 1: Image Caption Result 1

3 Implementation

1. From the `RNN_Captioning.ipynb` in Jupyter notebook, we implement the forward and backward pass for a vanilla RNN, first, 1) for a single timestep and then, 2) for entire sequences of data. Code to

check gradients has already been provided.

We overfit a captioning model on a tiny dataset and implement sampling from the softmax distribution and visualize predictions on the training and validation sets. The formula we used are shown below:

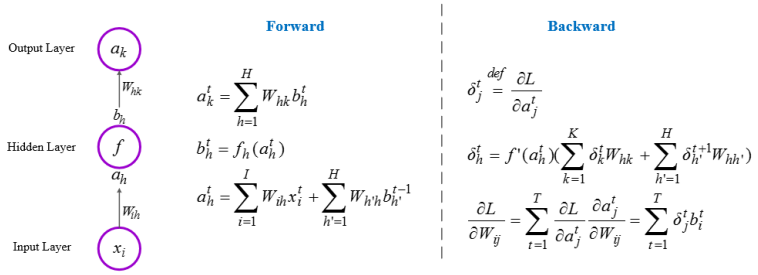

Figure 2: Forward and Backward Pass for a Vanilla RNN.

2. For a Recurrent Neural Network (RNN), a figure explaining its basic principle is shown below:

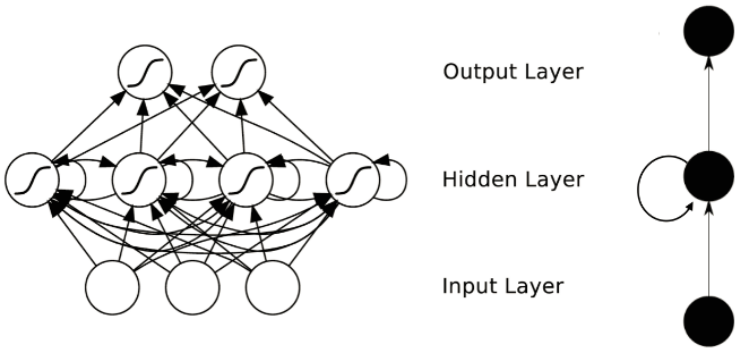

Figure 3: RNNs

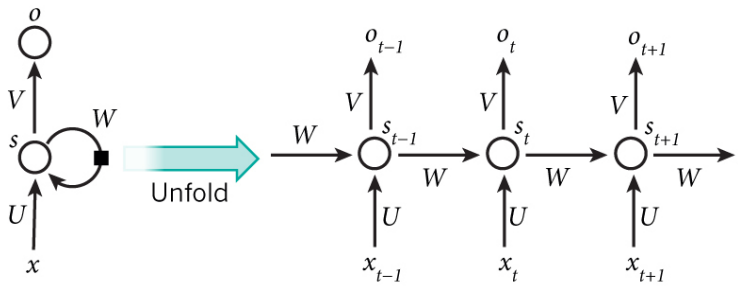

Figure 4: A RNN and the unfolding in time of the computation involved in its forward computation.

3. From the `LSTM_Captioning.ipynb` Jupyter notebook, we implement Long-Short Term Memory (LSTM) RNNs, and apply them to image captioning on MS-COCO.

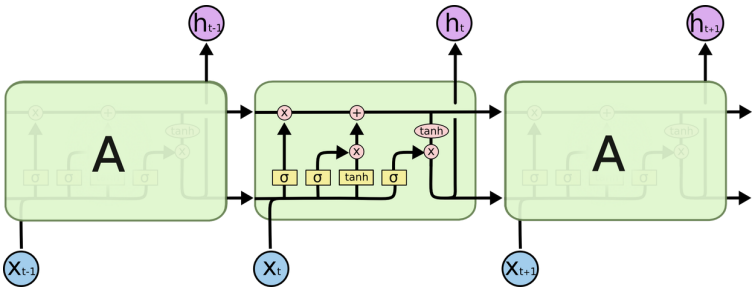

Figure 5: LSTM

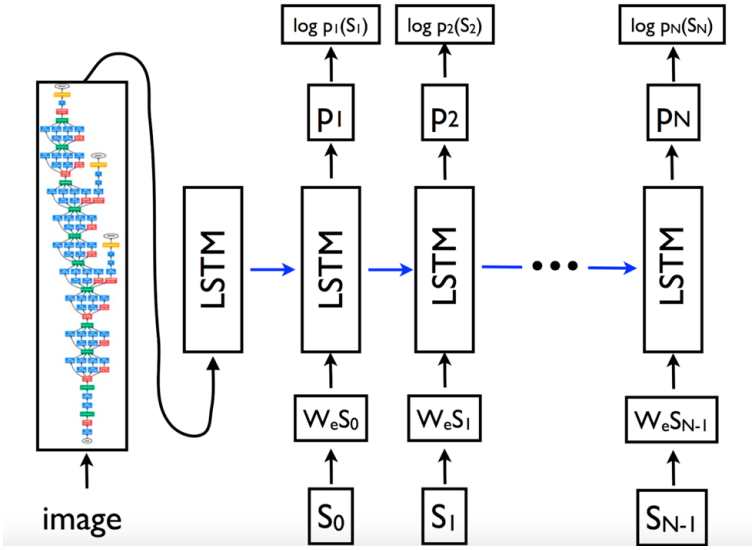

Figure 6: Long Short-Term Memory (LSTM) Network Decoder

4. Using the pieces we implement in parts 1 and 2, we train a captioning model that gives decent qualitative results (better than the random garbage you saw with the overfit models) when sampling on the validation set.

4 Results

5 References

1. Automatic Image Captioning using Deep Learning (CNN and LSTM) in PyTorch. `https://www.analyticsvidhya.com/blog/2018/04/solving-an-image-captioning-task-using-deep-learning/`.

2. `https://github.com/tensorflow/models/tree/master/research/im2txt`.

3. CS231n Convolutional Neural Networks for Visual Recognition. `http://cs231n.github.io/assignments2017/assignment3/`.

4. Attention-based captioning models. *[1, 7] Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. Xu et al., 2015.* `https://arxiv.org/abs/1502.03044`.

5. *Knowing When to Look: Adaptive Attention via A Visual Sentinel for Image Captioning. Lu et al., CVPR 2017.* `https://arxiv.org/abs/1612.01887`.

6. Discriminative captioning. *Context-aware Captions from Context-agnostic Supervision. Vedantam et al., CVPR 2017.* https://arxiv.org/abs/1701.02870.

7. Novel object captioning. *Deep Compositional Captioning: Describing Novel Object Categories without Paired Training Data. Hendricks et al., CVPR 2016.* https://arxiv.org/abs/1701.02870.

8. *Captioning Images with Diverse Objects. Venugopalan et al., CVPR 2017.* https://arxiv.org/abs/1701.02870.