EECS 498-007 / 598-005
Deep Learning for Computer Vision
Winter 2022

Assignment 5

In this assignment, you will implement two different attention-based models, RNN and Transformers. The goals of this assignment are:

- Understand and implement recurrent neural networks
- See how recurrent neural networks can be used for image captioning
- Understand how to augment recurrent neural networks with attention
- Understand and implement different building blocks of the Transformer model
- Use the Transformer model on a toy dataset

This assignment is due on **Tuesday, April 12th** at 11:59pm ET.

Q1: Image Captioning with Recurrent Neural Networks (50 points)

The notebook **rnn_lstm_attention_captioning.ipynb** will walk you through the implementation of vanilla recurrent neural networks (RNN) and Long Short Term Memory (LSTM) RNNs. You will use these networks to train an image captioning model. You will then augment your implementation to perform spatial attention over image regions while generating captions.

Q2: Transformer model for simple arithmetic operations (50 points)

The notebook **Transformers.ipynb** will walk you through the implementation of the Transformer model. You will first implement different building blocks referenced from the paper Attention is All You Need(https://arxiv.org/abs/1706.03762), and then use it on a toy dataset for simple addition and subtraction arithmetic expression. In the end, you will see what does a Transformer model learn by visualizing it's attention weights.

Steps

1. Download the zipped assignment file

- Click here to download the starter code

2. Unzip all and open the Colab file from the Drive

Once you unzip the downloaded content, please upload the folder to your Google Drive. Then, open each `*.ipynb` notebook file with Google Colab by right-clicking the *.ipynb file. We recommend editing your `*.py` file on Google Colab, set the ipython notebook and the code side by side. For more information on using Colab, please see our Colab tutorial.

3. Work on the assignment

Work through the notebook, executing cells and writing code in `*.py`, as indicated. You can save your work, both `*.ipynb` and `*.py`, in Google Drive (click "File" -> "Save") and resume later if you don't want to complete it all at once.

While working on the assignment, keep the following in mind:

- The notebook and the python file have clearly marked blocks where you are expected to write code. **Do not write or modify any code outside of these blocks**.
- **Do not add or delete cells from the notebook**. You may add new cells to perform scratch computations, but you should delete them before submitting your work.
- **IMPORTANT: Run all cells, and do not clear out the outputs, before submitting.** You will only get credit for code that has been run.

4. Evaluate your implementation on Autograder

Once you want to evaluate your implementation, please submit the `*.py`, `*.ipynb` and other required files to Autograder for grading your implementations in the middle or after implementing everything. You can partially grade some of the files in the middle, but please make sure that this also reduces the daily submission quota. Please check our Autograder tutorial for details.

5. Download .zip file

Once you have completed a notebook, download the completed `uniqueid_umid_A5.zip` file, which is generated from your last cell of the `transformers.ipynb` file. Before executing the last cell in `rnn_lstm_attention_captioning.ipynb`, please manually **run all the cells of notebook and save your results** so that the zip file includes all updates.

Make sure your downloaded zip file includes your most up-to-date edits; the zip file should include:

- *transformers.py*
- *rnn_lstm_attention_captioning.py*
- *Transformers.ipynb*
- *rnn_lstm_attention_captioning.ipynb*
- *rnn_lstm_attention_submission.pt*

- *transformer.pt*

6. Submit your python and ipython notebook files to Autograder

When you are done, please upload your work to Autograder (UMich enrolled students only). Your `*.ipynb` files *SHOULD include* all the outputs. Please check your outputs up to date before submitting yours to Autograder.

EECS 498-007 / 598-005: Deep Learning for Computer Vision

Justin Johnson
justincj@umich.edu

Website for UMich EECS course