

Assignment 07: Building Recurrent Neural Network Layers by Hand(RNN)

CSci 560: Neural Networks and Deep Learning

Description

Welcome to our first assignment over Text/Sequence deep learning systems. In this assignment you will implement by hand the key components of Recurrent Neural layers in NumPy.

Recurrent Neural Networks (RNN) are very effective for Natural Language Processing and other sequence tasks because they have “memory”. They can read inputs $x^{(t)}$ (such as words) one at a time, and remember some information/context through the hidden layer activations that get passed from one time-step to the next. This allows a unidirectional RNN to take information from the past to process later inputs. A bidirectional RNN can take context from both the past and the future.

Instructions:

- As with the previous assignment, you will need to create the function declarations asked for in `src/assg_tasks.py`. Make sure you use [Python Docstrings](#) and are generally following [Pep8 Python Style Guide](#) for your code.
- Cells with `### TESTED` comment contain unit tests that are run on your implementation. You will need to uncomment the call to the unit tests, but otherwise need to stay as given in the original notebook.
- Likewise since you need to write your declaration of the functions asked for the tasks, don't forget to uncomment/add the appropriate `from assg_src include X` statements in both this notebook and in the `../src/test_assg_tasks.py`

Objectives

You will learn:

- Implement the basic building blocks of a recurrent NN layer implementation.
- Learn more about the modifications of an LSTM, and adding its residual connections to avoid vanishing gradients issues.
- Learn in detail the operations of recurrent layer cells and how they work.
- See some examples of how recurrent layer operations can be unrolled in order to calculate gradients over the tensor operations performed by them.

Overview and Setup

Assignment Tasks

Assignment Submission

Additional Information