LING 574 HW7

Due 11:59PM on May 22, 2025

In this assignment, you will

- Develop an intuition of the flow of recurrent Seq2Seq models
- Implement key pieces of the model architecture
- Better understand the formal process of attention
- Learn and practice writing useful Python docstrings

All files referenced herein may be found in `/mnt/dropbox/24-25/574/hw7/` on patas.

1 Understanding Attention [20 pts]

Q1: Padding and Softmax [3 pts]

- What is $e^{-\infty}$? Or more accurately, what is $\lim_{x \to -\infty} e^x$?
- Give the equation for softmax over an input vector x for the value x_i
- What would happen to the softmax output if every x_i in x were set to $-\infty$?

Q2: Attention Calculation Complete the computation of attention given the following values for Queries and Keys. Interpret the Queries and Keys as having a hidden dimension of 2, with the other dimension being sequence length. Following the case of RNN Seq2Seq, Keys and Values are the same. You can use a library like Numpy or Pytorch to do the computations, but make sure to do them step by step. When filling in the matrices, round to two decimal places, but do not round in your actual calculations. [10 pts]

$$\underset{\text{Queries}}{\begin{bmatrix} 0.2 & 0.4 \\ 0.8 & 0.6 \\ 1.0 & 1.2 \end{bmatrix}} \times \underset{\text{Keys}^T}{\begin{bmatrix} 0.1 & 0.7 & 0.9 & 1.5 \\ 0.3 & 0.5 & 1.1 & 1.3 \end{bmatrix}} = \underset{\text{Compatibility}}{\begin{bmatrix} - & - & - & - \\ - & - & - & - \\ - & - & - & - \end{bmatrix}}$$

$$+ \underset{\text{Mask}}{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\infty \\ 0 & 0 & -\infty & -\infty \end{bmatrix}} = \underset{\text{Masked Compatibility}}{\begin{bmatrix} - & - & - & - \\ - & - & - & - \\ - & - & - & - \end{bmatrix}} \xrightarrow{\text{softmax}} \underset{\text{Attention}}{\begin{bmatrix} - & - & - & - \\ - & - & - & - \\ - & - & - & - \end{bmatrix}}$$

$$\times \underset{\text{Values}}{\begin{bmatrix} 0.1 & 0.3 \\ 0.7 & 0.5 \\ 0.9 & 1.1 \\ 1.5 & 1.3 \end{bmatrix}} = \underset{\text{Output}}{\begin{bmatrix} - & - \\ - & - \\ - & - \end{bmatrix}}$$

Q3: Attention Questions [7 pts]

- Give the equation for attention (i.e., the Output of Q2) in matrix notation, using Q, K, and V to represent the Queries, Keys, and Values matrices (no need to show actual numbers).
- In the Compatibility matrix, what does each row represent?
- Why do Keys and Values have to be the same length?
- In a recurrent encoder-decoder model, where do the Queries, Keys, and Values come from respectively?

2 Documenting Code [10 pts]

Docstrings are an important part of Python development, and so are an important part of developing a Deep Learning system. Use these links as a reference for Google style docstrings, which we will use here `https://sphinxcontrib-napoleon.readthedocs.io/en/latest/example_google.html`, `https://google.github.io/styleguide/pyguide.html#s3.8.1-comments-in-doc-strings`. You can also use the docstrings in `model.py` as examples. This format has the advantage of being able to be used to automatically generate html documentation from code for a project's website. Here, you will provide docstrings for the `Seq2SeqDataset` class in `data.py`, as well as its methods `batch_as_tensors` and `from_files`. Please only include the main description, `Args`, and `Returns`, giving roughly the level of detail in `model.py` (as opposed to the link).

The following section of the document is an environment that allows you to type monospaced plain-text in Latex. If you are using Latex, please make sure to keep your docstrings in these environments. If you are submitting a .txt file, you can just use `"""` to demarcate the beginning and end of your docstrings. This environment will wrap your lines for you, but try practicing manually formatting your lines to be 80 characters. Most text editors will allow you to show a ruler line at a specific number of characters, so try drafting your docstrings there and pasting them here.

```
class Seq2SeqDataset(Dataset):
"""
TODO: replace with your docstring here. Do NOT include the code.
"""

def batch_as_tensors(self, start: int, end: int) -> dict[str, Any]:
"""
"""

@classmethod
def from_files(
    cls, source_file: str, target_file: str, vocab: Vocabulary = None
):
"""
"""
```

3 Implementing a Recurrent Seq2Seq Model [37 pts]

In the remainder, you will implement key components of a character-level Seq2Seq translation model, based on LSTMs. The dataset we are using is called Europarl, and consists of proceedings of the EU parliament. The data files can be found at `/mnt/dropbox/24-25/574/data/europarl-v7-en-es/`. We are translating English to Spanish. All of the methods that you will be implementing are in `model.py`. **Read all docstrings and comments closely for desired behavior**.

Q1: Seq2SeqModel.forward Implement the `forward` method for the `Seq2SeqModel` class. This method should take in batches of input and target sequences, embed them, encode the input sequence and then use its final states and attention to decode the target sequence. **This method will call Seq2SeqModel.encode and Seq2SeqModel.decode**. [9 pts]

Q2: Seq2SeqModel.encode Implement the `encode` method for the module. This method should take in the batch of input sequences and their lengths, and return the position-wise hidden states as well as the final hidden and cell states. [9 pts]

Q3: Seq2SeqModel.decode Implement the `decode` method for the module. This method should take in a batch of target sequences, plus the hidden and final states from the encoder, and use the decoder LSTM + attention to convert to logits over the vocabulary for the target sequence. **This method will call Seq2SeqModel.attention** [9 pts]

Q4: Seq2SeqModel.attention Implement the `attention` module. This method should take series of vectors acting as Queries, as well as those acting as Keys/Values, plus a padding mask that we will provide for you. **The padding mask is created for you in forward, and must be passed to this method**. [10 pts]

4 Running the Translation Model [8 pts]

`run.py` contains a training loop to use the Seq2Seq translation model, which will record the training loss and generate text every N epochs (controlled by `--generate_every`, set to 1 by default). **Warning: the run of this model will take several hours or more. It is important that you plan ahead, and only use the default parameters of the script (which are small), until you are sure your code runs end to end without errors and it is time to run your final script.**

Q1: Run with Full Parameters Execute `run.py` with the following (full) arguments. Paste below the texts that are generated every epoch, as well as the training and validation loss. In 2-3 sentences, describe any trends that you see. [4 pts]

- `--train_source /mnt/dropbox/24-25/574/data/europarl-v7-es-en/train.en.txt`
- `--train_target /mnt/dropbox/24-25/574/data/europarl-v7-es-en/train.es.txt`
- `--output_file test.en.txt.es`
- `--num_epochs 8`
- `--embedding_dim 16`
- `--hidden_dim 64`
- `--num_layers 2`

Q2: Evaluate Translations Finally, we will evaluate the translations using Character F-Score (`https://www.aclweb.org/anthology/W15-3049/`). Using the provided script `chrF++.py`, calculate the score of the output Spanish translations using the following command, and report the score for the line that says `c6+w0-F2`: [4 pts]

```
python chrF++.py -nw 0 \
    -R /mnt/dropbox/24-25/574/data/europarl-v7-es-en/test.es.txt \
    -H test.en.txt.es > test.en.txt.es.score
```

5 Testing your code

In the dropbox folder for this assignment, you will find a file `test_all.py` with a few very simple unit tests for the methods that you need to implement. You can verify that your code passes the tests by running `pytest` from your code's directory, with the course's conda environment activated.

Submission Instructions

In your submission, include the following:

- readme.(txt|pdf) that includes your answers to §1, §2, and §4.
- `hw7.tar.gz` containing:
  - run_hw7.sh. This should contain the code for activating the conda environment and your run commands for §4 above. You can use run_hwX.sh from any previous assignment as a template.
  - model.py
  - test.en.txt.es