Homework Assignment #2

The one with the RNNs

Description

This homework assignment focuses on developing your skills to work with recurrent neural networks.
It has two main tasks:

1. Implement a basic Vanilla RNN and understand the role of **teacher forcing, warm start** and **capturing of the structure of interest in a sequence** on hand of a simple **Sine Wave Dataset**.
2. To **develop, train and test** your own **sequence-to-sequence model** for a language translation problem

Task 1 [5p]

The objective of this task is the implementation of a simple RNN cell and to understand how teacher forcing and warm start affect the **prediction capabilities** of an RNN.

The Jupyter notebook attached to this assignment provides the setup for this task. The mentioned TODOs follow this schema:

- TODO 1.1 targets implementing one ***step()*** of the forward pass for the RNN
- TODO 1.2 implements the whole forward step of an RNN by going over the entire input sequence.
    - It asks to implement **teacher forcing during training** (input at next time step *t* is given by the ground truth at time step *t* instead of the prediction made by the network at time step *t-1*)
    - It asks you to implement **warm start during evaluation** - for a **limited amount of timesteps** the network is given ground truth input instead of its own predictions

After implementing the TODOs, you will train the network on the simple task of predicting values from a sine wave.
Several hyperparameters control the learning and generation of sine sequences.

- UNROLL_LENGTH - gives the length of subsequences from the sine wave used during training
- TEACHER_FORCING_PROB - controls the probability of applying teacher forcing
- WARM_START - sets the max number of timesteps considered in the warm starting of the RNN when in prediction mode
- NUM_ITERATIONS, LEARNING_RATE and REPORTING_INTERVAL - control the training and reporting process

In your training perform the following experiments:

- Run with **different values of teacher forcing probability (p=0.0, p=0.5, p=0.75, p=1.0)** using the default values for all other parameters
- Keep Teacher forcing set at 0.5 and use a **very small** and a **very large** UNROLL_LENGTH
- Use UNROLL_LENGTH = 62 (maximum), set teacher forcing **off (p=0.0)** and use a warm start of only 2.
- Use an UNROLL_LENGTH of 3, put teacher forcing **on (p=1.0)** and use a warm start of 2.

After performing the experiments answer to the following questions:

1. Difference between teacher forcing and learning on own samples:
    a. What are the pros and cons of teacher forcing?
2. In which setup (combination of unroll_length and teacher forcing probability) is the model struggling to learn?
3. How does warm starting affect test time prediction? Why?
4. What happens if the structure of interest is much longer than the unroll length?

Task 2 [5p]

This task focuses on the development of a sequence-to-sequence architecture using LSTM or GRU layers.

The typical architecture of a basic sequence-to-sequence model for *language translation* is shown in the Figure 1 below, which shows an unrolled representation of the *encoding* and *decoding* modules for a french-to-english translation of the phrase “Good evening”.

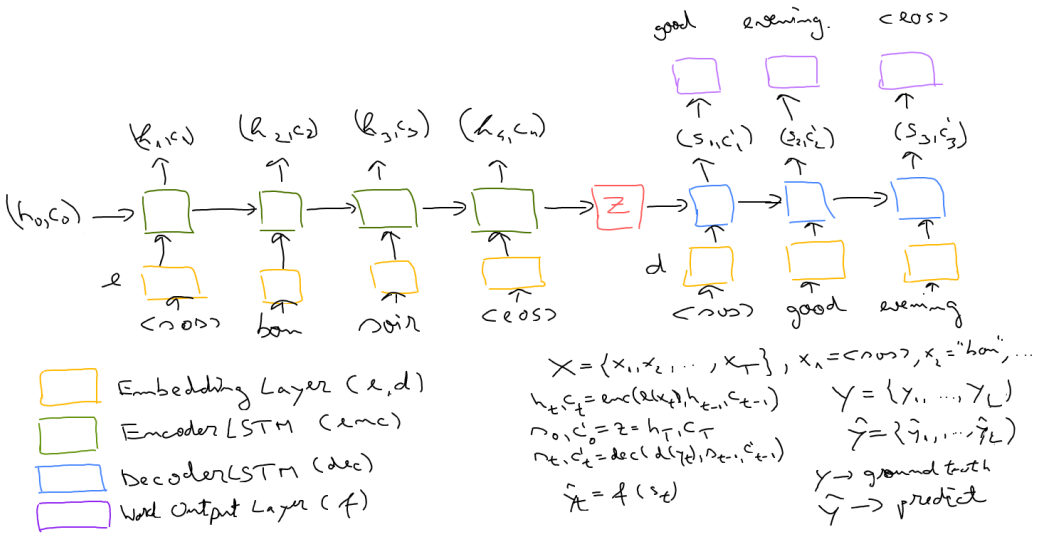

Figure 1. Depiction of a simple language translation seq-2-seq model using single layer LSTMs.

The network comprises text embedding layers (one for the encoder - french words - and one for the decoder - english words).
Encoder and Decoder are implemented as single layer LSTMs.
The Word Output Layer (purple depiction in Figure 1) converts the hidden state of the decoder ($\mathbf{s}_t$) to a vector the size of the output language vocabulary. It uses a Linear Layer to do this and applies a softmax on top of it to indicate the most likely token.

You have to perform the following subtasks.

The objective is to implement a seq-2-seq model for the french-to-english translation task.

To do so we use the Multi30k dataset. This is a dataset with ~30,000 parallel English, German and French sentences, each with ~12 words per sentence.
To obtain tokenizers for French and English sentences, you can use the **spaCy** small language models for French ("fr_core_news_sm") and English ("en_core_web_sm"). See here an example documentation for how to load and use the language models for tokenization.
The support code for the translation task provides you with methods to:

- Load the datasets from spaCy - splitting them into train and test subsets
- Obtain a vocabulary for English and French, which contains special tokens for **start-of-sentence (sos), end-of-sentence (eos), unknown (unk)** and **padding (pad)** tokens. The vocabulary allows you to convert between a sentence and its token-index representation and vice-versa.
- Implement collation, such that one can **pad** sentences of different length to a common size, useful when batching sentences.

Objective 1. Implement and test a **single layer LSTM encoder** and **decoder model** for a seq-to-seq translation model trained on the Multi30k dataset.
Experiment with:

- Using input dropout - adding a dropout layer (with probability 0.1 or 0.5) after the initial text embedding layer
- Different sizes of the word embedding layer (e.g. 128, 256, 512)
- Different sizes of the hidden dimension (e.g. 128, 256, 512)
- Different batch sizes (e.g. 128, 256)
- Different values of teacher forcing applied to the decoding process (e.g. p=0.0, p=0.5, p=1.0)

Objective 2. For the best configuration from Objective 1, modify the encoder architecture to use a **bidirectional LSTM model**, while the decoder remains unidirectional.

For each experiment from both objectives you are required to:

- Provide training and test loss curves to analyze model training
- Provide **BLEU score** and **Perplexity** values highlighting the performance of the model. If your code uses a PyTorch Lightning setup, you may find the torchmetrics package useful for computing the metrics for perplexity and BLEU score
- Discuss the key findings for each experiment, highlighting the setups that performed better and providing your justification for the performance.