CS 563: Natural Language Processing
Assignment-4: Neural Language Model

Deadline: 29 April 2023,

- Markings will be based on the correctness and soundness of the outputs.
- Marks will be deducted in case of plagiarism.
- Proper indentation and appropriate comments (if necessary) are mandatory.
- Use of frameworks like scikit-learn, PyTorch etc is allowed.
- *All benchmarks(accuracy etc), answers to questions and supporting examples should be added in a separate file with the name 'report'*.
- *All code needs to be submitted in '.py' format.* Even if you code it in '.IPYNB' format, download it in '.py' format and then submit
- You should zip all the required files and name the zip file as:
    - <roll_no>_assignment_<#>.zip, eg. 1501cs11_assignment_01.zip.
- Upload your assignment (the zip file) in the following link:
    - https://www.dropbox.com/request/ZiYE4PhK7L5mDhIMgWya

Problem Statement:

- The assignment targets to implement 2-gram and 3-gram character-level language models with Feed Forward Neural Network

Dataset:

- Names dataset:
    - Dataset consists of the most common 32K names taken from ssa.gov for the year 2018.
    - Link: https://www.dropbox.com/s/6vnpqv5cacgljs0/names.txt?dl=0
    - **Example:**
        - zhiheng
        - ziaan
        - zichen
        - zidon

Implementation:

- Pre-process the data and append full stop at the end of every name
- Input to the network consist of character n-grams
- The model is trained to predict next character given input n-gram

- For example, if the input is “zidon.” then the training set consists of following pairs (if the input is 2-gram)
    - zi -> d
    - id -> o
    - do -> n
    - on -> .
- The vocabulary consists of every unique character in the dataset

Input to the Neural Network:

- Input to the NN should be one-hot encoding of input tokens (similar to Assignment-3)
- For example, given the following name:

    zidon.

- Vocabulary size: **27** (26 characters and full stop)
- The one-hot encoding for the characters is as follows:

    z: **[1, 0, 0, 0, 0, 0, 0, 0, …, 0]**
    i: **[0, 1, 0, 0, 0, 0, 0, 0, …, 0]**
    …

- The dimensionality of input: [2 x 27] (in case of 2-gram) or [3 x 27] (in case of 3-gram)
- Note: You can also introduce a batch dimension
- Since the network takes a fixed length input (2 in case of 2-gram and 3 in case of 3-gram), no need to PAD the corpus.

Feed-Forward NN:

- Explain and draw the architecture of Feed-Forward NN that you are proposing with justification. Describe the features of Feed-Forward NN.
- Network should contain **TWO** hidden layers
    - input — hidden_layer_1 (hidden_layer_1 size is 128)
    - hidden_layer_1 — hidden_layer_2 (hidden_layer_2 size is 64)
- Finally, hidden_layer_2 — Output (Output size is 27 as the model needs to predict any one of the character from the vocabulary)
- Use non-linearity of your choice (tanh, relu, gelu etc.) between hidden layers

Evaluation:

- Split the dataset and use 90% as trainset 5% as devset and remaining 5% as testset
- Run each model for 20 epochs (minimum)
- Save best model checkpoint based on the **Perplexity of dev set**
- Report **Perplexity** for the best model checkpoint on **test** set
- Note: **Perplexity = e^(loss)** where the loss is calculated with cross-entropy loss function

Documents to submit:

- Model code
- Model logs (in the form of graph):
    - Perplexity of dev set for each epoch
    - Train loss for each epoch
- Write a report (doc or pdf format) on how you are solving the problems as well as all the results including model architecture (if any).

For any queries regarding this assignment, contact:

Gopendra Singh Vikram (gopendra.99@gmail.com),
Mamta (mamta20118@gmail.com),
Aizan Zafar (aizanzafar@gmail.com),
Ramakrishna Appicharla (ramakrishnaappicharla@gmail.com) and,
Arpan Phukan (arpanphukan@gmail.com)