TRINITY COLLEGE DUBLIN

School of Computer Science and Statistics

Week 9 Assignment CS7CS4/CSU44061 Machine Learning

Rules of the game:

• Its ok to discuss with others, but do not show any code you write to others. You must write answers in your own words and write code entirely yourself. All submissions will be checked for plagiarism.

• Reports must be typed (no handwritten answers please) and submitted as a separate pdf on Blackboard (not as part of a zip file please).

• Important: For each problem, your primary aim is to articulate that you understand what you're doing - not just running a program and quoting numbers it outputs. Long rambling answers and “brain dumps" are not the way to achieve this. If you write code to carry out a calculation you need to discuss/explain what that code does, and if you present numerical results you need to discuss their interpretation. Generally most of the credit is given for the explanation/analysis as opposed to the code/numerical answer. Saying “see code" is not good enough, even if code contains comments. Similarly, standalone numbers or plots without further comment is not good enough.

• When your answer includes a plot be sure to (i) label the axes, (ii) make sure all the text (including axes labels/ticks) is large enough to be clearly legible and (iii) explain in text what the plot shows.

• Include the source of code written for the assignment as an appendix in your submitted pdf report. Also include a separate zip file containing the executable code and any data files needed. Programs should be running code written in Python, and should load data etc when run so that we can unzip your submission and just directly run it to check

that it works. Keep code brief and clean with meaningful variable names etc.

• Reports should typically be about 5 pages, with 10 pages the upper limit (excluding appendix with code). If you go over 10 pages then the extra pages will not be marked.

• Note: In this assignment there is no need to use cross-validation to select hyperparameters - deep learning needs too much computing power for that to be practical.

Assignment

In this assignment you'll take a closer look at transformers.

(i) Download the file week9Assignment.zip from Blackboard. Unzip it into an empty folder. Ensure that you can run the script gpt.py on your machine. The code outputs some generated text that gives you an idea of what the model learnt (and did not learn). Note that there are three datasets in the folder. A training and test datasets; and a different dataset from Shakespeare’s plays.

   a. Modify the code so that you load the *input_childSpeech_trainingSet.txt* dataset. Briefly describe the dataset (e.g., what does it appear to contain? What is the vocabulary size? What is the length?). Do the same for the other two datasets in the folder.
   b. Load the *input_childSpeech_trainingSet.txt* dataset. Downsize the model configuration so that it has less than 1 million parameters by manipulating the hyperparameters on lines 5-16 of *gpt.py*. Describe the rationale for your downsizing (i.e., why did you downsize a particular parameter rather than another one).
   c. Attempt another two different ways of downsizing and report the loss functions for the three configurations. Discuss your results in terms of validation loss, overfitting, and qualitative assessment of the generated output. You can constrain your execution to 1000 iterations, if 5000 are too computational expensive for your machine.
   d. Explore and describe in the report how the inclusion of the bias terms in the self-attention layers impacts the transformer model.

e. Explore and describe in the report how the inclusion of skip connections throughout the transformer architecture impacts the resulting model.

(ii) Next, you will evaluate the model on the two test sets *input_childSpeech_testSet.txt* and *input_shakespeare.txt*. That will require adding some code to load and evaluate your model on test sets.

a. Select the best model from part (i). Calculate the test loss on *input_childSpeech_testSet.txt.* Report it and comment on it. Is it good, bad? Why? How does it compare to a dummy/baseline model?

b. Calculate the loss on *input_shakespeare.txt*. Report it and comment on it. Is it good, bad? Why? How does it compare to a dummy/baseline model? How does it compare with the loss for question (ii)(a)? How do we interpret this result? How could this pipeline be used in practice?

The error message indicates that the character 'F' is not found in the stoi dictionary, which means it was not present in the training text. To handle this, you can modify the encode function to skip characters that are not in the stoi dictionary or map them to a special token.

Here's a step-by-step plan to fix this issue:

Add a special token for unknown characters.
Modify the encode function to handle unknown characters.
Let's implement these changes:

Add a special token for unknown characters: