Deep Learning:
Assignment 3: Machine Translation

Instructor: Jia Xu

Fall 2020

Due: December 15th 2020

Adversarial Machine Translation Inputs (10 Points)

Google translate is one of the most successful machine translation systems in the world. The use of deep learning has significantly advanced translation quality. However, the translation can still output errors given arbitrary inputs. Your task is to find and collect as many as possible foreign input sentences that Google translate will make errors on.

- step 1: Open Google translate
- step 2: try different inputs in any language that will make the translation output errors
- step 3: list those adversarial input examples in a table: left column: input sentence; middle column: machine translation output that is erroneous; right column: correct translation output. (Your score will be given based on the number of different hacking sentences, the errors each sentence caused, and the diversity of the sentences.) [6 points]
- step 4: analyze what types of errors and possible reasons for those errors [2 point]
- step 5: what are your ideas of improving neural networks to avoid those errors? [2 point]

Bonus Question: Attentional Neural Machine Translation (10 Points – this is optional!)

You should train a neural network model to automatically translate from English words to their transformated forms. The transformation rules are as follows:

- if the first letter is a consonant, then that letter is moved to the end of the word and "ay" is appended, e.g., slow $\rightarrow$ lowsay
- if a word starts with a vowel, then append "way" at the end, e.g., amoeba $\rightarrow$ amoebaway.

- some consonant pairs like "sh" are moved together to end of the word with "ay" appended, e.g., shallow $\rightarrow$ allowshay.

To translate a sentence, simply translate each word independently.
Your tasks:

1. build and train a machine translation system using data.txt to learn these rules implicitly. You should use a character based translation model. That means each word token is a character.

2. The first model you use is based on RNN, sequence to sequence model. The encoder RNN compresses the input sequence into a fixed-length vector, represented by the final hidden state h_T . The decoder RNN conditions on this vector to produce the translation, character by character. Input characters are passed through an embedding layer before they are fed into the encoder RNN; in our model, we learn a 2910 embedding matrix, where each of the 29 characters in the vocabulary is assigned a 10-dimensional embedding. At each time step, the decoder RNN outputs a vector of unnormalized log probabilities given by a linear transformation of the decoder hidden state. When these probabilities are normalized, they define a distribution over the vocabulary, indicating the most probable characters for that time step. The model is trained via a cross-entropy loss between the decoder distribution and ground-truth at each time step. [2 Points]

3. Split the data.txt into 80% training and 20% and get the accuracy on the test set. [2 Points]

4. Question: If you input a very long sentence, do you have problem of translation this sentence? If so, what could be potentially the problem? Write down the answer. [2 Points]

5. What model or activation function can you use the improve long sentence translation? Implement such a model. [2 Points]

6. Design and implement an attention model that reduces the error you get from the test set. [2 Points]

In data.txt, we group word pairs based on the lengths. So, in each mini-batch the source words and target words are all the same length, respectively, you do not have to worry about batches of variable-length sequences.

Assignment 3 Submission Guidelines:

1. Adversarial Machine Translation Inputs - Submit a single pdf containing your solution. If the pdf contains handwritten solutions, make sure it is legible.

2. Bonus Question: Attentional Neural Machine Translation - Submit working code and a pdf report containing training and test accuracies, answers to the question from task 4, and any other information that will be required to understand your solution.