CS224 Winter 2019 Assignment 5

Name: Dat Nguyen

Date: 3/14/2019

By turning in this assignment, I agree by the Stanford honor code and declare that all of this is my own work.

1. Character-based convolutional encoder for NMT (36 points)

(a) Because we do convolution over characters, we can still get high level of representation for words with just characters of low embedding size.

(b) Total number of parameters in character-based embedding model is

$$\begin{aligned} & e_{\text{char}} \times V_{\text{char}} + e_{\text{word}} \times e_{\text{char}} \times k + e_{\text{word}} + 2 \times e_{\text{word}} \times e_{\text{word}} + 2 \times e_{\text{word}} \\ =& e_{\text{char}} \times V_{\text{char}} + e_{\text{word}} \times e_{\text{char}} \times k + 2 \times e_{\text{word}} \times e_{\text{word}} + 3 \times e_{\text{word}} \end{aligned}$$

Total number of parameters in word_based lookup embedding model is

$$V_{\text{word}} \times e_{\text{word}}$$

From there we get total parameters for character-based embedding model is 200640 which is 63 times few than 12800000 parameters of word_based look up embedding model.

(c) In NMT task we use word embedding ultimately to encode and decode sentences but not to generate new word. RNN might be inappropriate because it tries to model the order relationship between characters in a word but that relationship may not be necessary to have a good word representation for the upper task. CNN learns the local structure of a word so it might more easily learn good word representation to adapt to the NMT task.

(d) In max-pooling the most activated window will be recorded for each filter, so each filter can learn the local structure in a sharper, less noisy way. On the other hand, average-pooling takes into consideration many windows so it may retain more information than max-pooling.

(h) To check that my implementation of highway network is correct, I printed out and verified the shape of all intermediate layers. In addition, I manually input the weights and biases of $\mathbf{W}_{\text{proj}}$ and $\mathbf{W}_{\text{gate}}$ and compared the final result with the result I got by doing the computation step by step in numpy. I also asserted that the shape of the final output was correct.

(i) To check that my implementation of CNN is correct, I printed out and verified the shape of all intermediate layers. In addition, I manually input the weights and biases of **W** and compared the final result with the result I got by doing the computation step by step in numpy. I also asserted that the shape of the final output was correct.

2.Character-based LSTM decoder for NMT (26 points)

(f) My BLEU score: 24.47

3. Analyzing NMT Systems (8 points)

(a) Of these sis forms, "traducir" and "traduce" appear while "traduzco", "traduces", "traduzca" and "traduzcas" do not appear in the vocabulary. The problem with word-based NMT is that when encoding, the word which does not appear in the vocabulary is treated as UNKNOWN so that causes us lose some semantic information, therefore in decoding we might not be able to output the corresponding word in the target language. The character-aware NMT recognizes a word by the its characters so we do not lose information about that word in encoding. In addition, even if the target vocabulary does not have the appropriate word, character-aware NMT can still output that word because it might have encountered the similar situation in the training process.

(b) i Most nearest neighbors for words trained by Word2Vec

- financial: economic
- neuron: nerve
- Francisco: san
- naturally: occurring
- expectation: norms

ii Most nearest neighbors for words trained by CharCNN

- financial: vertical
- neuron: Newton
- Francisco: France
- naturally: practically
- expectation: exception

iii The similarity modeled by Word2Vec is semantic similarity while similarity between sequence of characters is modeled by CharCNN. Because of the way Word2Vec is trained, frequently occur together words will have low cosine distance, therefore we may expect that similar words will be semantically close. On the other hand, CharCNN does convolution over adjacent characters so it is likely that words having similar sequence of characters will be close.

(c) i
- Source sentence: Mi crculo comenz en los aos '60 en la escuela media, en Stow, Ohio donde yo era el raro de la clase
- Reference translation: My circle began back in the '60s in high school in Stow, Ohio where I was the class queer.
- Word-based translation: My circle started in the year <unk >at the middle school, in Ohio – where I was the rare of the class.
- Char-based translation: My circle started in the '60s in the middle school, in Stanford, Ohio.
- Explanation: This is an acceptable example. In training the system has probably seen similar pattern "" followed by numbers so it can decode successfully.

ii
- Source sentence: Qu opinaremos del hecho de diferir de ellos en slo unos pocos nucletidos?
- Reference translation: What are we to make of the fact that we differ from them only really by a few nucleotides?
- Word-based translation: What do you find? Of them in just a few <unk >
- Char-based translation: What will we forget about the fact that they are in just a few nuclears?
- Explanation: This is a not acceptable example because "nucletidos" in the Spanish source sentence is translated wrongly as "nuclears". Because words embedding are learned by CNN, in training the system probably see words having similar characters as "nucletidos" translated as "nuclears" so it adapts to the newly seen "nucletidos" this way.