Stanford CS 224n Assignment 5

Shivanshu Shekhar

December 2021

1 Model description and written questions

(a) Word embedding are very high dimensional as there are many words in any language and also the same word can have multiple meaning so the embedding should capture all these details, whereas the character of any language are few in number and also a single word doesn't hold any meaning generally so we don't need as much parameters to represent character as we need for words.

(b)

Number of parameters in character level model:

$$total\_params = parmas_{embedding} + params_{CNN} + params_{Highway}$$
$$total\_params = V_{char} \cdot e_{char} + (e_{char} \cdot f \cdot k + e_{char}) + 2 \cdot (e_{word}\dot{e}_{word} + e_{word})$$

Number of parameters in word level model:

$$total_params = params_{embedding}$$
$$total_params = e_{word} \cdot V_{vocab}$$

$Fork = 5, V_{word} = 50,000$ and $V_{char} = 96, e_{char} = 50, e_{word} = 256$

We get for both the models:

$$Char\_level\_model = 200,640$$
$$Word\_level\_model = 12,800,000$$

(c) Deep CNN models is easy to make and train where as we cant make much deep RNN models as RNN is slower to compute because it calculates output sequentially so more parameters can be achieved by using CNN rather than RNN which is good for learning better representations.

(d) Maxpooling tells us about which feature got the most activated for an input which can be better in understanting the content of the input where as Avgpooling can potentially ignore the high activation as it sums over all activation and lower activations can mask the effect of high activations.

Avgpooling preserves all features of the input where as Maxpooling ignores other features in favour of the highest activation.

2 Implementation

(e) CODE

(f) CODE

(g) CODE

(h) I performed the following tests:

- I checked the size of the output tensor by passing a single tensor input generated randomly
- I also passed two self designed tensor in a batch format to check if its output matches with what i have calculated.
- I also check some special cases that I though will have a high probability of failing.

I think that these tests were enough as the sizes of tensors are the first thing that will cause an error as these tensors are supposed to have a fixed predefined shape, next I ran a demo test and matched the values with what I calculated, this I did was because as these error wont show any runtime error but the model will still not be able to train properly, I also checked with some special cases which I thought the model could fail.

(i) Same as above

(j) CODE

(k) CODE

(l) CODE

3 Character-based LSTM decoder for NMT (26 points)

The BLEU scores are not possible to report as I dont have enough resources to run the model in full capacity but the small local tests were passed and I got 99.24 BLEU score on that.

I will report the new BLEU scores when I get access to the necessary resources.

4 Analyzing NMT Systems (8 points)

This sections couldnt be completed as I could'nt trainb the model fully due to GPU restrictions = (.