cs224n

(arthurg)

June 5, 2020

Question 1.

When using convolutions on character level language models, the convolutions (of size k) are able to operate on words of arbitrary length. However, the output of this convolution will be a vector of size $len - k$.

Question 2.

The minimum w_{word} is 1. After padding the *sow* and *eow* tokens, the minimum length for x_{padded} would thus be $R^{1+2} = R^3$

To ensure we apply at least one full convolution, we need to padd x_{padded} to size 5. Then, we need padding of 1. $x_{reshaped} \in R^{e_{char} x 1+2+(2*1)} = R^{e_{char} x 5}$

Question 3.

It's useful for the extremes of x_{gate} to set $x_{highway}$ be either fully x_{proj} or fully x_{conv_out} because it allows certain character embeddings to optionally pass through another layer.

It's probably a better idea to set the bias to positive. This will ensure $x_{gate} - > 1$. If $x_{gate} = 0$, we will have no gradient on x_{proj} which makes the layer useless.

Question 4.

- Parallizes better on GPUs

- Multi-headed attention might improve translation accuracy when trying to do things like verb - noun agreements

Question 5.

I tested my Highway network by:

1. Running a batch of x_{conv_out} to check the dimensions are correct

2. Set the weights of w, and send in an x_{conv_out} vector (of size 5). Manually do the matrix math to check the output matches. In these test cases, I made sure to find cases where the sigmoid is 0, sigmoid is 1, and where the relu is 0.

I'm confident that these two tests will cover the edge cases. In general, since I used pre-defined components in PyTorch, most of the edge cases are handled for me.

Question 6.

I tested my CNN network by:

1. Running the CNN on a batch of x_{emb} to check the dimensions are correct

2. Set the weights of w, and send in an x_{emb}. Through hand computing the convlustion, and pooling operation, I manually network was acting properly.

3. Set the weights of w, and send in an x_{emb}. Through hand computing the convlustion, and RELU operation, I manually network was acting properly.

I'm confident that these three tests will cover the edge cases. In general, since I used pre-defined components in PyTorch, most of the edge cases are handled for me.

Question 7.

My BLEU score is 36.51893424356271

Question 8.

traducir -> Not in vocab

traduzco -> In vocab

traduces -> Not in vocab

traduce -> In vocab

traduzca -> In vocab

traduzcas -> Not in vocab

This would be bad for the word-based NMT because the encoder would be unable to a few conjugations of 'traducir'. If the unseen versions of the verb occurs in the test set, the system likely output an <UNK> word.

The character-based NMT model overcomes this as it would likely learn an embedding for the common substring, namely 'tradu'. Then, if it sees a conjugation of the verb that is not in the vocab, it uses the prefix of the word to estimate an embedding, and then this prefix will most likely accuratley encode the meaning.

Question 9.

Financial -> Economic

Neuron -> Nerve

Francisco -> San

naturally -> occuring

expectation -> norms

Question 10.

Financial -> Vertical

Neuron -> Newton

Francisco -> France

naturally -> practically

expectation -> exception

Question 11.

In the Word2Vec embeddings, it appears that the neighbours have similar meaning or are words that are used in conjunction with one another.

In the CharCNN embeddings, it appears that the neighbours have either a similar prefix or suffix. Most of the time, the neighbours don't have any relation to one another semantically.

Question 12.

Incorrect example first

Source: Hoy estoy aqu para hablarles sobre crculos y epifanas.

Reference : I'm here today to talk to you about circles and epiphanies.

Output (A4): I'm here to talk to you about circles and <unk>

Output (A5): I'm here today to talk to you about circles and epidemiologists.

The UNK got translated as epidemiologists.

During this translation, i think the encoder encountered 'epifanas' in spanish which it never saw before. So, it probably used the CharCNN model, which means that the word had similar embedding to 'epidemióloga' (meaning epiodemiologist). Then, the decoder probably had the word 'epidemiologist' in its vocab, and outputed that.

Correct Example"

Source: Entonces ella me deca: "Bueno y qu hars cuando vengas?"

Reference: And she'd say, "Well what are you going to do when you get here?

Output (A4) : So she said, "Well what are you going to do when you <unk>

Output (A5) : So she said,“Well, what do you do when you come? ”

The UNK got translated as 'get here'

During this translation, I think the encoder encountered 'vengas', which was probably not in the spanish vocab for the encoder. As a result, in A4, this was translated as UNK. In a5, I believe the CharCNN model encoder was probably able to find a match for the root word of 'vengas' (perhaps venir?). Then, the word was decoded properly.