Part I CSC421 PA3 2019年05月30日

1. The architecture performance should deteriorate. If the input length is N, the input becomes further and further away until it is used, making the signal weak.
eg. Consider the consonant rule. The first consonant becomes ~2N away from when it is used. The rest ~N.

2. From Sutskever et. al:
- reverse the input sequence but keep the output sequence
- use LSTMs which can store long-term memory

3. At train time, teacher forcing means the gradient signal that should flow from the output through the input and through the output of the previous step is not used.

At train time, each output is conditioned on the true previous word, but at test time, it is actually a RV with an uncertainty. This uncertainty is not modelled, thus, the RNN is overly confident.

At train time teacher forcing can perhaps "reset" the error, if there is an error. At test time, the error would accumulate.

4. Using the terminology of the paper, the difference b/w train and test is that at train, $h_t = P(h_{t-1}, y_{t-1}, \theta)$ while at test, $h_t = P(h_{t-1}, \hat{y}_{t-1}, \theta)$ where y_{t-1} is the true prev. token and $\hat{y}_{t-1}$ is the estimate.
The paper proposes that at train time, the conditioning on the previous token should be a choice b/w the true value and the model estimate, and this choice is determined by a binomially distributed RV, ϵ_i. If $\epsilon_i = 1$, use true, else.
During the start of training the prob. that $\epsilon_i = 1$ should be high to help the model learn, but later, it should be low, to mimic test time & allow it to explore.

Part 3

The output after 99 epochs was:

```
Epoch:  99 | Train loss: 0.757 | Val loss: 1.107 | Gen: ehay airypay
ontingivelay isway ordelingway
source: the air conditioning is working
translated: ehay airypay ontingivelay isway ordelingway
```

The correct output is "ethay airway onditioningcay isway orkingway." The NN is capturing some of the rules of pig-latin such as moving the first letter of consonant starting words forward. But as can be seen, it failed to capture the rule for blocks such as 'th' and failed for longer words.

The model was also used to translate the following sentence:

```
source: this string should be relatively difficult to explain to
anthropologists
translated: isthay ingstray ouldshay ebay elortielyway iffiductway otay
expialmay otay orostostingowcay
```

Correct: isthay ingstray ouldshay ebay elativelyray ifficultday otay explainway otay anthropologistsway

Again, it is clear that the model fails on long words.

The training time was 630s.

Part 4

NN Based (Additive) Attention

After 99 epochs the final result was:

```
Epoch:  99 | Train loss: 0.001 | Val loss: 0.069 | Gen: ethay airway
onditioningcay isway orkingway
source: the air conditioning is working
translated: ethay airway onditioningcay isway orkingway
```

The example sentence was translated perfectly. Below is another example sentence

```
source: this string should be relatively difficult to explain to
anthropologists
translated: isthay ingstray ouldshay ebay elativelyray ifficultday otay
explainway otay anthropologistsway
```

As can be seen, the translator got every single word correct, including the three-letter-block-containing-word "string."

The training time was ~490s, which is comparable to the non-attention based RNN, but this is a confusing result because there are more connections (between each encoder embedding and the decoder).

Scaled Dot Product

```
Epoch:  99 | Train loss: 0.007 | Val loss: 0.108 | Gen: ethay airway
onditioniongcay isway orkingway
source: the air conditioning is working
translated: ethay airway onditioniongcay isway orkingway

source: this string should be relatively difficult to explain to
anthropologists
translated: isthay ingstray ouldshay ebay elativelyray ifficultday otay
explainway otay anthrolosistsway
```

The model failed to correctly translate the longest word 'anthropologists'. There was a higher training and validation loss. The degradation in accuracy can be explained because the new f function is simple a linear mapping, whereas before a neural network was used which had higher capacity due to non-linear activation functions.

The training time was 1143s. This is confusing because one might expect it to be faster than the additive attention model because this model can calculate all the attention coefficients simultaneously via matrix multiplcation on a GPU instead of passing each key and query through a neural network.

Part 5 - Transformers

1. The additive attention method should be slower because of the lack of parallelization when computing attention coefficients and context vectors but it has a higher capacity and it can achieve better results. The scaled dot product method is the opposite.

3. After 99 epochs the output was:

```
Epoch:  99 | Train loss: 0.003 | Val loss: 0.068 | Gen: ethay airway
onditiongcay isway orkingway
source: the air conditioning is working
translated: ethay airway onditiongcay isway orkingway

source: this string should be relatively difficult to explain to
anthropologists
translated: isthay ingstray ouldshay ebay elativelyray ifficultday otay
explainway otay anthrologististmay
```

The model was able to correctly translate the test sentence except for one mistake translating 'anthropologists.' It outperformed the scaled dot product based attention model and it achieved comparable performance with the NN-based attention model.

The training time was 235s, faster than the previous models including the NN-based attention model. This speed-up is due to the parallel nature of the transform model, whereas RNN outputs cannot be computed in parallel due to their recursive nature.

4.

```
Epoch:  99 | Train loss: 0.296 | Val loss: 0.366 | Gen: eay ayEOSy y
isaysaysasasasasasas orky
source:          the air conditioning is working
translated: eay ayEOSy y isaysaysasasasasasas orky
```

As expected, the performance was very poor, probably because the model used the future outputs to cheat during training.

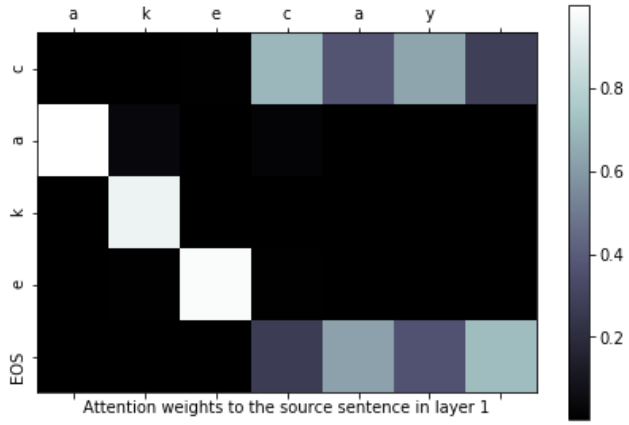
5. I would guess that due to the causality mask, the model learns to attend to the first few inputs in the early stages. Additionally, it can detect blocks or consonants and correspondingly do a shift to attend to the true first letter that needs to be placed. As more outputs get generated, that somehow moves the attention to the encoded states.

Part 6

RNN (Additive) Attention Map

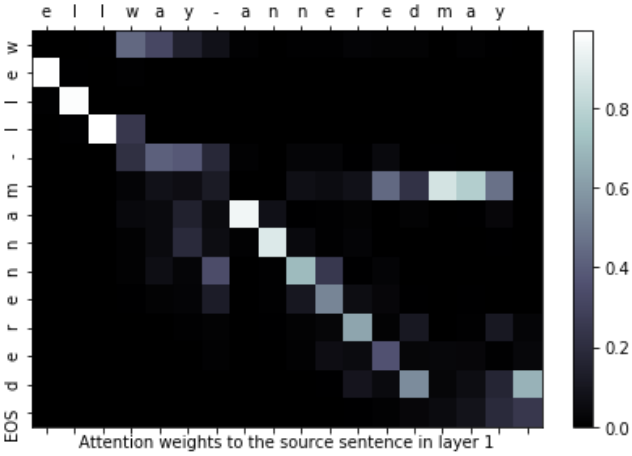
Cake

The attention map for 'cake' shows that 'a', 'k', 'e' are all highly attentive to themselves which is what one would expect, and 'c', 'a', 'y' are highly attentive to itself and the EOS, which makes sense because they "wait" until the word is finished.



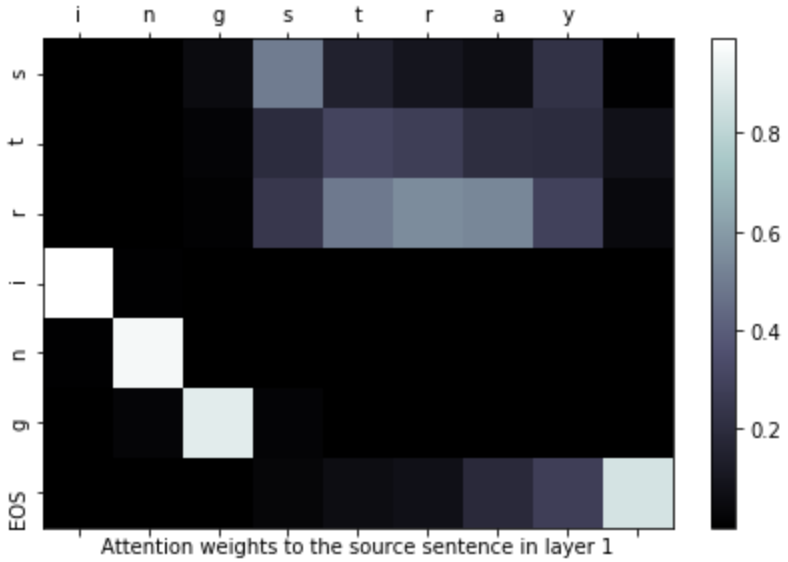
Well-Mannered

The interesting aspect of the map for 'well-mannered' is that the model is able to correctly separate the two hyphenated words such that one sees along the diagonal the pattern of the map for cake twice along the diagonal.



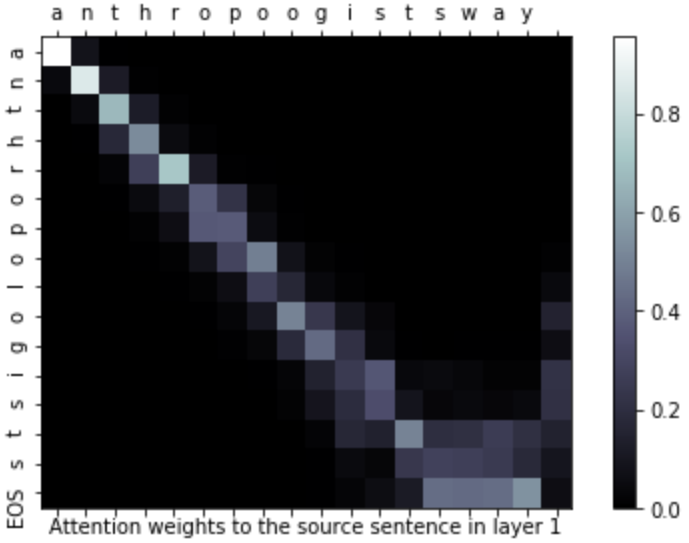
String

Impressively, when outputing 'str', the neural network treats 'str' as a block as can be seen in the heatmap below.



Anthropologists

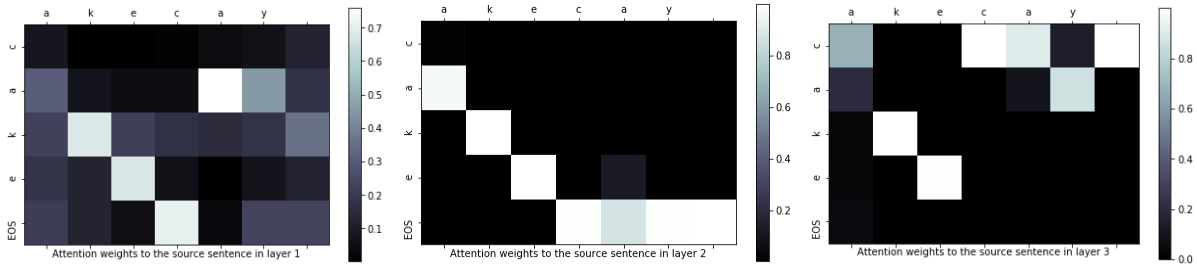
Although it is unclear why, it is apparent that as the word becomes longer, the model looks at a larger surrounding local context, as seen below by the diagonal "expanding." This then causes the model to fail.



Transformer Attention Map

Cake

Interestingly, the second layer and third layer combined somewhat mimics the attention map of the RNN; the second layer focuses on outputting the beginning of the word as well as being attentive to the EOS character when outputting the final characters, and in the third layer the final few characters of the output is highly attentive to the 'c'.



Anthropologists

Although it is unclear why, as the model came to the center of the word 'anthropologists', it suddenly spread its attention and made a mistake. But then it re-attends to the correct places as it continues the output.