Encoder (BoW)

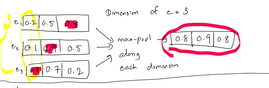
I like apple.

Bow-sum $\Rightarrow e_1 + e_2 + e_3 \Rightarrow c$

Bow-mean $\Rightarrow \frac{e_1 + e_2 + e_3}{3} \Rightarrow c$

Bow-maxpool $\rightarrow$ maxpool $(e_1, e_2, e_3) \Rightarrow c$

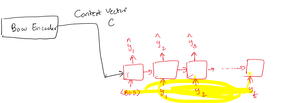
Dimension of $c \rightarrow 3$



Decoder

Keep decoding step by step
→ until we find $\langle EOS \rangle$ token
→ or ~~we've hit target sentence length~~.

How would we train the decoder?



Feed real target words to decoder instead of previous output.

Teacher forcing

How do we evaluate translation?

Reference: I like llama, a ~~cute animal that lives~~ in South America.

Prediction: ~~cute animal that lives~~

Modified n-gram precision: $P_n = \frac{\sum_{sec} \sum_{gram \in S} \hat{C}(\text{ngram})}{\sum_{sec} \sum_{gram \in S} C_{ref}(\text{ngram})}$

$\hat{C}(\text{ngram}) = \min(C(\text{ngram}), C_{ref}(\text{ngram}))$

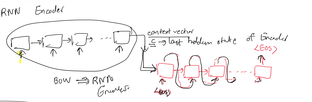
$P_1 = \frac{1 + 1 + 1 + 1}{1 + 1 + 1 + 1} = 1$

$P_4 = \frac{1}{1} = 1$

You can just predict a short phrase and get maximum prediction!!! ⇒ NOT GOOD

~~Brevity penalty~~ ⇒ $BP = \begin{cases} 1 & \text{if } L > r \\ \exp(1 - \frac{r}{L}) & \text{if } L \le r \end{cases}$

RNN Encoder



BOW ⇒ RNN Encoder

What if we only want to look at the input words necessary for translating the next word?

⇒ Attention mechanism

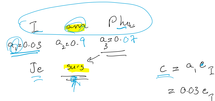
→ Not fixed context

h_i is hidden state of source sentence.

keys and values.

Context vector C is the weighted sum of encoder hidden states.

If weight a_i for word i is low, the influence of word i on output is low.



$c = a_1 e_I + a_2 e_{am} + a_3 e_{Phu}$

$= 0.03 e_I + \overset{0.9}{a_2} e_{am} + \overset{0.07}{a_3} e_{Phu}$

high-max

~~high weight~~

How to get the weights?

To decode time step t, the previous decoder hidden s_{t-1} called query.

Find score $= f(s_{t-1}, h_j)$ for word j in encoder

f can be neural net $\Rightarrow f(s_{t-1}, h_j) = \text{FFNN}(s_{t-1}, h_j)$

f can be ~~just → dot-product~~ ⇒

scaled dot-product ⇒ $\frac{s_{t-1} \cdot h_j}{\sqrt{d}}$ ← (dimension of h)

$a_{tj} = \text{softmax}(f(s_{t-1}, h_j))$

query → q key → k

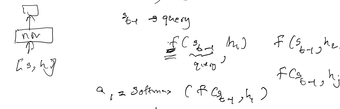
$c_t = a_{1t} h_1 + a_{2t} h_2 + \cdots + a_{Lt} h_L$

value → v

$= \sum_{i=1}^{L} \text{softmax}(f(q_t, k_i)) \cdot v_i$

At time t, we want to

$s_{t-1} \rightarrow$ query



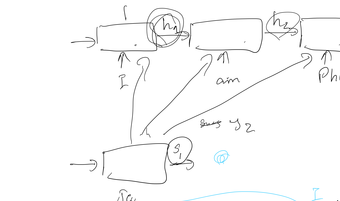
$f(s_{t-1}, h_1)$ $f(s_{t-1}, h_2)$ ---

$f(s_{t-1}, h_j)$

$a_j = \text{softmax}(f(s_{t-1}, h_j))$

$c = a_1 h_1 + a_2 h_2 \cdots + a_j h_j$

$s^T h$



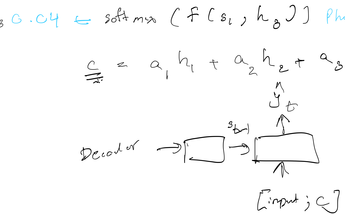
Je Sus Phu

a_1 0.05 ← softmax $(f(s_1, h_1))$ → dot product $(s_1^T h_1)$

a_2 0.9 ← softmax $(f(s_1, h_2))$ am $(s_1^T h_2)$

a_3 0.04 ← softmax $(f(s_1, h_3))$ Phu $(s_1^T h_3)$

$c = a_1 h_1 + a_2 h_2 + a_3 h$



$\text{softmax}(f(q, k)) \cdot v$

s_{t-1} h h