NMT Model.

from embedding: $x_1, \cdots, x_m$. $R^{e \times 1}$

embedding
$\downarrow$

CNN

$h_i^{enc} = [\overrightarrow{h_i^{enc}}, \overleftarrow{h_i^{enc}}]$ $R^{e \times 1}$
$c_i^{enc} = [\overrightarrow{c_i^{enc}}, \overleftarrow{c_i^{enc}}]$

$\downarrow$

bidirectional encoder $R^{2h \times 1}$

$\downarrow$

decoder

$h_0^{dec} = W_h [\overrightarrow{h_m^{enc}}, \overleftarrow{h_1^{enc}}]$ $R^{h \times 1}$
$c_0^{dec} = W_c [\overrightarrow{c_m^{enc}}, \overleftarrow{c_1^{enc}}]$

$h_t^{dec}, c_t^{dec} = \text{Decoder}(\overline{y_t}, h_{t-1}^{dec}, c_{t-1}^{dec})$ $\downarrow$

$e_{t,i} = (h_t^{dec})^T W_{attProj} h_i^{enc}$ $R^{m \times 1}$

$\alpha_t = \text{softmax}(e_t)$ $\downarrow$

$a_t = \sum_{i=1}^{M} \alpha_{t,i} h_i^{enc}$ $R^{2h \times 1}$

$\downarrow$

$u_t = [h_t^{dec}; a_t]$ $R^{3h \times 1}$

$v_t = W_u u_t$ $R^{h \times 1}$

$o_t = \text{dropout}(\tanh(v_t))$ $R^{h \times 1}$

probability distribution $P_t = \text{softmax}(W_{vocab} o_t)$ $R^{V_t \times 1}$

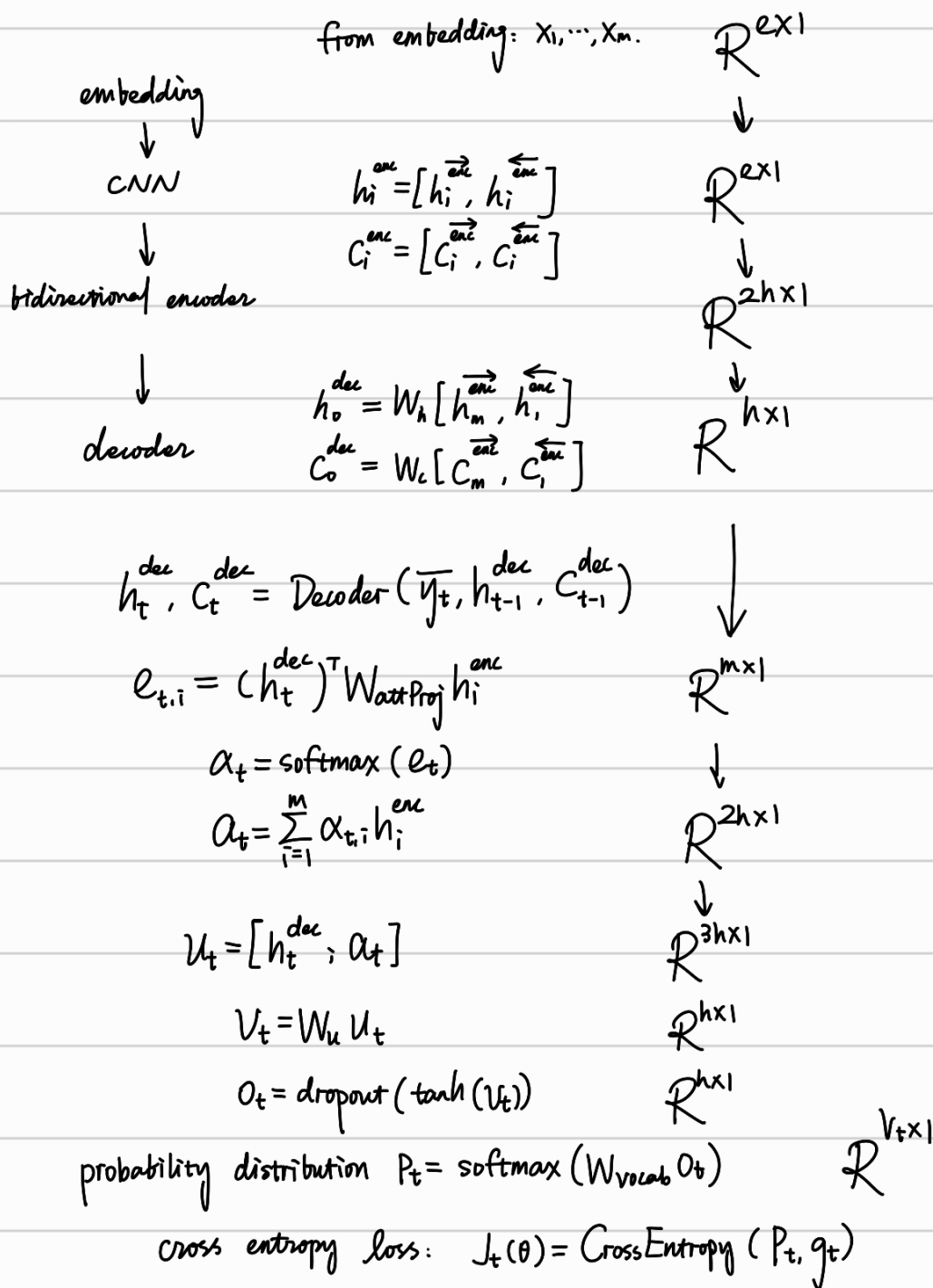cross entropy loss: $J_t(\theta) = \text{CrossEntropy}(P_t, g_t)$

1. (g) According to the masks, set the masked part of e_t to $-\infty$, so that part is neglected after calculating α_t because of the softmax function. Consequently, the latter part of sentence or the padding won't interfere with the current generating process.

(h) BLEU = 18.35 > 18

(i)

(i). dot product attention $e_{t,i} = S_t^T h_i$ v.s.   multiplicative attention $e_{t,i} = S_t^T W h_i$.

dot product : pro: easier to calculate $\Rightarrow$ same complexity, but dot product can be highly parallelized

① S_t and h_i need to be identically shaped ② gradient vanishing

con: ~~not utilizing~~ so much information. (i.e. ~~S_t and h_{i2} are not directly connected~~)

(ii).

additive attention $e_{t,i} = V^T \tanh(W_1 h_i + W_2 S_t)$ v.s.   multiplicative attention $e_{t,i} = S_t^T W h_i$.

additive attention : pro: less parameters, better alignment ability

con: more complicated calculation

2. (a) Considering there are plenty of morphemes in Chinese characters, adding a 1D Convolutional layer helps to connect these characters. For example, combining adjacent "电" "脑" to get "电脑" so that the information of words are carefully extracted.

(b) i. NMT failed to use the plural from of "culprit", because "妖人" might be used to refer to a single person or a group, which is not clearly indicated in the sentence.

ii. NMT repeated itself and neglected half of the original sentence.

Maybe add hidden layer size to fix it.

iii. NMT left out an important part of the sentence.

Maybe add vocab size so that rarely used words are considered as well.

iv. NMT failed to generate an ancient-style sentence.

Maybe add some ancient Chinese sentences to the training dataset.

(c)(i.) BLEU score for C_1: $p_1 = \frac{4}{9}$, $p_2 = \frac{3}{8}$, $BP = \exp(1 - \frac{11}{9})$

$BLEU_1 = \exp(1 - \frac{11}{9}) \times \exp(0.5 \times \frac{4}{9} + 0.5 \times \frac{3}{8}) = 1.206$

BLEU score for C_2: $p_1 = 1$, $p_2 = \frac{3}{5}$, $BP = 1$

$BLEU_2 = \exp(1 - \frac{6}{5}) \times \exp(0.5 \times 1 + 0.5 \times \frac{3}{5}) = 2.226$

$BLEU_1 < BLEU_2$, but I think C_1 is better than C_2

(ii) $p_1 = \frac{4}{9}$, $p_2 = \frac{3}{8}$. $BP = 1$. $BLEU_1 = \exp(\frac{2}{9} + \frac{3}{16}) = 1.506$

$p_1 = \frac{1}{2}$, $p_2 = \frac{1}{5}$. $BP = 1$. $BLEU_2 = \exp(\frac{1}{4} + \frac{1}{10}) = 1.419$

$BLEU_1 > BLEU_2$ and I do think C_1 is better than C_2.

(iii) ① can't recognize synonyms

② can't recognize correct sentence structures if they aren't identical to the only given reference

③ can't evaluate its grammar correctness thoroughly

(iv) advantage: automatic evaluation (fast, steady)

disadvantage: requires a comprehensive reference database

can't recognize complicated grammar structure.