A Detailed Architecture

This appendix describes in detail the implementation of the *self-attentive residual decoder* for NMT, which builds on the attention-based NMT implementation of `dl4mt-tutorial`[1].

The input of the model is a source sentence denoted as 1-of-k coded vector, where each element of the sequence corresponds to a word:

$$x = (x_1, x_2, ..., x_m), x_i \in \mathbb{R}^V$$

and the output is a target sentence denoted as well as 1-of-k coded vector:

$$y = (y_1, y_2, ..., y_n), y_i \in \mathbb{R}^V$$

where V is the size of the vocabulary of target and source side, m and n are the lengths of the source and target sentences respectively. We omit the bias vectors for simplicity.

A.1 Encoder

Each word of the source sentence is embedded in a e-dimensional vector space using the embedding matrix $\bar{E} \in \mathbb{R}^{e \times V}$. The hidden states are $2d$-dimensional vectors modeled by a bi-directional GRU. The forward states $\overrightarrow{h} = (\overrightarrow{h}_1, ..., \overrightarrow{h}_m)$ are computed as:

$$\overrightarrow{h}_i = \overrightarrow{z}_i \odot \overrightarrow{h}_{i-1} + (1 - \overrightarrow{z}_i) \odot \overrightarrow{h}'_i$$

where

$$\begin{aligned}
\overrightarrow{h}'_i &= tanh(\overrightarrow{W}\bar{E}x_i + \overrightarrow{U}[\overrightarrow{r}_i \odot \overrightarrow{h}_{i-1}]) \\
\overrightarrow{z}_i &= \sigma(\overrightarrow{W}_z\bar{E}x_i + \overrightarrow{U}_z\overrightarrow{h}_{i-1}) \\
\overrightarrow{r}_i &= \sigma(\overrightarrow{W}_r\bar{E}x_i + \overrightarrow{U}_r\overrightarrow{h}_{i-1})
\end{aligned}$$

Here, $\overrightarrow{W}, \overrightarrow{W}_z, \overrightarrow{W}_r \in \mathbb{R}^{d \times e}$ and $\overrightarrow{U}, \overrightarrow{U}_z, \overrightarrow{U}_r \in \mathbb{R}^{d \times d}$ are weight matrices. The backward states $\overleftarrow{h} = (\overleftarrow{h}_1, ..., \overleftarrow{h}_m)$ are computed in similar manner. The embedding matrix $\bar{E}$ is shared for both passes, and the final hidden states are formed by the concatenation of them:

$$h_i = \begin{bmatrix} \overrightarrow{h}_i \\ \overleftarrow{h}_i \end{bmatrix}$$

[1] `https://github.com/nyu-dl/dl4mt-tutorial`

A.2 Attention Mechanism

The *context vector* at time t is calculated by:

$$c_t = \sum_{i=1}^{m} \alpha_i^t h_i$$

where

$$\begin{aligned}
\alpha_i^t &= \frac{exp(e_i^t)}{\sum_j exp(e_j^t)} \\
e_i^t &= v_a^\intercal tanh(W_d s_{t-1} + W_e h_i)
\end{aligned}$$

Here, $v_a \in \mathbb{R}^d$, $W_d \in \mathbb{R}^{d \times d}$ and $W_e \in \mathbb{R}^{d \times 2d}$ are weight matrices.

A.3 Decoder

The input of the decoder are the previous word y_{t-1} and the *context vector* c_t, the objective is to predict y_t. The hidden states of the decoder $s = (s_1, ..., s_n)$ are initialized with the mean of the *context vectors*:

$$s_0 = tanh(W_{init} \frac{1}{m} \sum_{i=1}^{m} c_i)$$

where $W_{init} \in \mathbb{R}^{d \times 2d}$ is a weight matrix, m is the size of the source sentence. The following hidden states are calculated with a GRU conditioned over the *context vector* at tine t as follows:

$$s_t = z_t \odot s'_t + (1 - z_t) \odot s''_t$$

where

$$\begin{aligned}
s''_t &= tanh(Ey_{t-1} + U[r_t \odot s_{t-1}] + Cc_t) \\
z_i &= \sigma(W_z E y_{t-1} + U_z s_{t-1} + C_z c_t) \\
r_i &= \sigma(W_r E y_{t-1} + U_r s_{t-1} + C_r c_t)
\end{aligned}$$

Here, $E \in \mathbb{R}^{e \times V}$ is the embedding matrix for the target language. $W, W_z, W_r \in \mathbb{R}^{d \times e}$, $U, U_z, U_r \in \mathbb{R}^{d \times d}$, and $C, C_z, C_r \in \mathbb{R}^{d \times 2d}$ are weight matrices. The intermediate vector s'_t is calculated from a simple GRU:

$$s'_t = GRU(y_{t-1}, s_{t-1})$$

In the attention-based NMT model, the probability of a target word y_t is given by:

$$\begin{aligned}
p(y_t | s_t, y_{t-1}, c_t) = softmax(W_o tanh(& \\
W_{st} s_t + W_{yt} y_{t-1} + W_{ct} c_t))&
\end{aligned}$$

Here, $W_o \in \mathbb{R}^{V \times e}$, $W_{st} \in \mathbb{R}^{e \times d}$, $W_{yt} \in \mathbb{R}^{e \times e}$, $W_{ct} \in \mathbb{R}^{e \times 2d}$ are weight matrices.

A.3.1 Self-Attentive Residual Connections

In our model, the probability of a target word y_t is given by:

$$p(y_t|s_t, d_t, c_t) = softmax(W_o tanh(\\ W_{st}s_t + W_{dt}d_t + W_{ct}c_t))$$

Here, $W_o \in \mathbb{R}^{V \times e}$, $W_{st} \in \mathbb{R}^{e \times d}$, $W_{dt}, W_{yt} \in \mathbb{R}^{e \times e}$, $W_{ct} \in \mathbb{R}^{e \times 2d}$ are weight matrices. The summary vector d_t can be calculated in different manners based on previous words y_1 to y_{t-1}. First, a simple average:

$$d_t^{avg} = \frac{1}{t-1} \sum_{i=1}^{t-1} y_i$$

The second, by using an attention mechanism:

$$d_t^{cavg} = \sum_{i=1}^{t-1} \alpha_i^t y_i$$

$$\alpha_i^t = \frac{exp(e_i^t)}{\sum_{j=1}^{t-1} exp(e_j^t)}$$

$$e_i^t = v^\intercal tanh(W_y y_i)$$

where $v \in \mathbb{R}^e$, $W_y \in \mathbb{R}^{e \times e}$ are weight matrices.

A.3.2 Memory RNN

This model modifies the recurrent layer of the decoder as follows:

$$s_t = z_t \odot s'_t + (1 - z_t) \odot s''_t$$

where

$$s''_t = tanh(Ey_{t-1} + U[r_t \odot \tilde{s}_t] + Cc_t)$$

$$z_i = \sigma(W_z E y_{t-1} + U_z \tilde{s}_t + C_z c_t)$$

$$r_i = \sigma(W_r E y_{t-1} + U_r \tilde{s}_t + C_r c_t)$$

Here, $E \in \mathbb{R}^{e \times V}$ is the embedding matrix for the target language. $W, W_z, W_r \in \mathbb{R}^{d \times e}$, $U, U_z, U_r \in \mathbb{R}^{d \times d}$, and $C, C_z, C_r \in \mathbb{R}^{d \times 2d}$ are weight matrices. The intermediate vector s'_t is calculated from a simple GRU:

$$s'_t = GRU(y_{t-1}, \tilde{s}_t)$$

The recurrent vector $\tilde{s}_t$ is calculated as following:

$$\tilde{s}_t = \sum_{i=1}^{t-1} \alpha_i^t s_i$$

where $$\alpha_i^t = \frac{exp(e_i^t)}{\sum_{j=1}^{t-1} exp(e_j^t)}$$

$$e_i^t = v^\intercal tanh(W_m s_i + W_s s_t)$$

where $v \in \mathbb{R}^d$, $W_m \in \mathbb{R}^{d \times d}$, and $W_s \in \mathbb{R}^{d \times d}$ are weight matrices.

A.3.3 Self-Attentive RNN

The formulation of this decoder is as following:

$$p(y_t|y_1, ..., y_{t-1}, c_t) \approx softmax(W_o tanh(\\ W_{st}s_t + W_{yt}y_{t-1} + W_{ct}c_t + W_{mt}\tilde{s}_t))$$

Here, $W_o \in \mathbb{R}^{V \times e}$, $W_{st} \in \mathbb{R}^{e \times d}$, $W_{yt} \in \mathbb{R}^{e \times e}$, $W_{ct} \in \mathbb{R}^{e \times 2d}$, and $W_{mt} \in \mathbb{R}^{e \times d}$ are weight matrices.

$$\tilde{s}_t = \sum_{i=1}^{t-1} \alpha_i^t s_i$$

$$\alpha_i^t = \frac{exp(e_i^t)}{\sum_{j=1}^{t-1} exp(e_j^t)}$$

$$e_i^t = v^\intercal tanh(W_m s_i + W_s s_t)$$

where $v \in \mathbb{R}^d$, $W_m \in \mathbb{R}^{d \times d}$, and $W_s \in \mathbb{R}^{d \times d}$ are weight matrices.