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.

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

\[ \hat{h}_i = \frac{\sum_j e^{j}}{\sum_j e^{j}} \sum_j e^{j} tanh(W_{d} s_{t-1} + W_{e} h_{i}) \]

Here, \( y_t \) and \( y_{t-1} \) are the previous word and target language. \( W_d, W_{e} \) are weight matrices. The backward states \( \hat{h} = (\hat{h}_1, ..., \hat{h}_m) \) are computed in similar manner. The embedding matrix \( E \) is shared for both passes, and the final hidden states are formed by the concatenation of them:

\[ h_i = [\hat{h}_i, \hat{h}_i^\top] \]

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

\[ \alpha_i^t = \frac{exp(\epsilon_i^t)}{\sum_j exp(\epsilon_j^t)} \]
\[ \epsilon_i^t = v^t_i tanh(W_{d} s_{t-1} + W_{e} h_{i}) \]

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 time \( t \) as follows:

\[ s_t = z_t \odot s_t' + (1 - z_t) \odot s_t'' \]

where

\[ s_t'' = tanh(Ey_{t-1} + Ut \odot st-1 + Ct) \]
\[ z_t = \sigma(W_z Ey_{t-1} + Ut s_{t-1} + Ct) \]
\[ r_t = \sigma(W_r Ey_{t-1} + Ut s_{t-1} + Ct) \]

Here, \( E \in \mathbb{R}^{e \times V} \) is the embedding matrix for the target language. \( W_z, W_r, W_{e} \in \mathbb{R}^{d \times e}, U_t, U_r \in \mathbb{R}^{d \times d}, \) 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:

\[ s_t' = GRU(y_{t-1}, st-1) \]

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

\[ p(y_t | s_t, y_{t-1}, c_t) = \text{softmax}(W_y tanh(W_s s_t + W_y y_{t-1} + W_c c_t)) \]

Here, \( W_y \in \mathbb{R}^{e \times e}, W_{st} \in \mathbb{R}^{e \times d}, W_{y} \in \mathbb{R}^{e \times e}, W_{ct} \in \mathbb{R}^{e \times 2d} \) are weight matrices.
A.3.1 Self-Attentive Residual Connections

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

$$p(y_t|s_t, d_t, c_t) = \text{softmax}(W_o \cdot \text{tanh}(W_st + W_dt d_t + W_c c_t))$$

Here, $W_o \in \mathbb{R}^{V \times e}$, $W_s \in \mathbb{R}^{e \times d}$, $W_dt, W_y \in \mathbb{R}^{e \times e}$, $W_c \in \mathbb{R}^{e \times 2d}$, and $W_m \in \mathbb{R}^{d \times d}$ 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:

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

where

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

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

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

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

$$z_t = \sigma(W_s Ey_{t-1} + U_s \tilde{s}_t + C_e c_t)$$

$$r_t = \sigma(W_r Ey_{t-1} + U_r \tilde{s}_t + C_r c_t)$$

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

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

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

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

where

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

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

where $v \in \mathbb{R}^d$, $W_m \in \mathbb{R}^{d \times a}$, 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:

$$p(y_t|y_1, ..., y_{t-1}, c_t) \approx \text{softmax}(W_o \cdot \text{tanh}(W_st + W_y y_{t-1} + W_c c_t + W_m \tilde{s}_t))$$

Here, $W_o \in \mathbb{R}^{V \times e}$, $W_s \in \mathbb{R}^{e \times d}$, $W_y \in \mathbb{R}^{e \times e}$, $W_c \in \mathbb{R}^{e \times 2d}$, and $W_m \in \mathbb{R}^{d \times d}$ are weight matrices.

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

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

$$e_i^t = v^t \tanh(W_m s_i + W_s \tilde{s}_t)$$

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