

C code implementation progress of Transformer-based TTS

Sian-Yi Chen

Advisors : Tay-Jyi Lin and Chingwei Yeh

Outline

Action item

- Transformer-based TTS C implementation

Status report

- 先前進度
 - ❑ 完成底層Linear layer多維矩陣的各種操作，維度交換、維度擴增、多維矩陣相乘等。
 - ❑ 完成至Encoder中的Multi-head attention layer。
- 本周進度
 - ❑ 已完成Encoder C code implementation
 - Encoder總共重複6次($N=6$)，共計48層，計算時重複使用相同陣列儲存，計算Encoder重複過程中，因為有些矩陣重複儲存時沒有初始化(歸0)，導致數值錯誤，而多花費了不少時間。
 - ❑ 目前正在整理Decoder的演算法
 - Decoder與Encoder不同的地方為多了一塊Masked Multi-head attention layer(紅框)，因為推理過程是產生未知語句，因此跟平行處理的Encoder不一樣是遞迴處理(一次一個字)

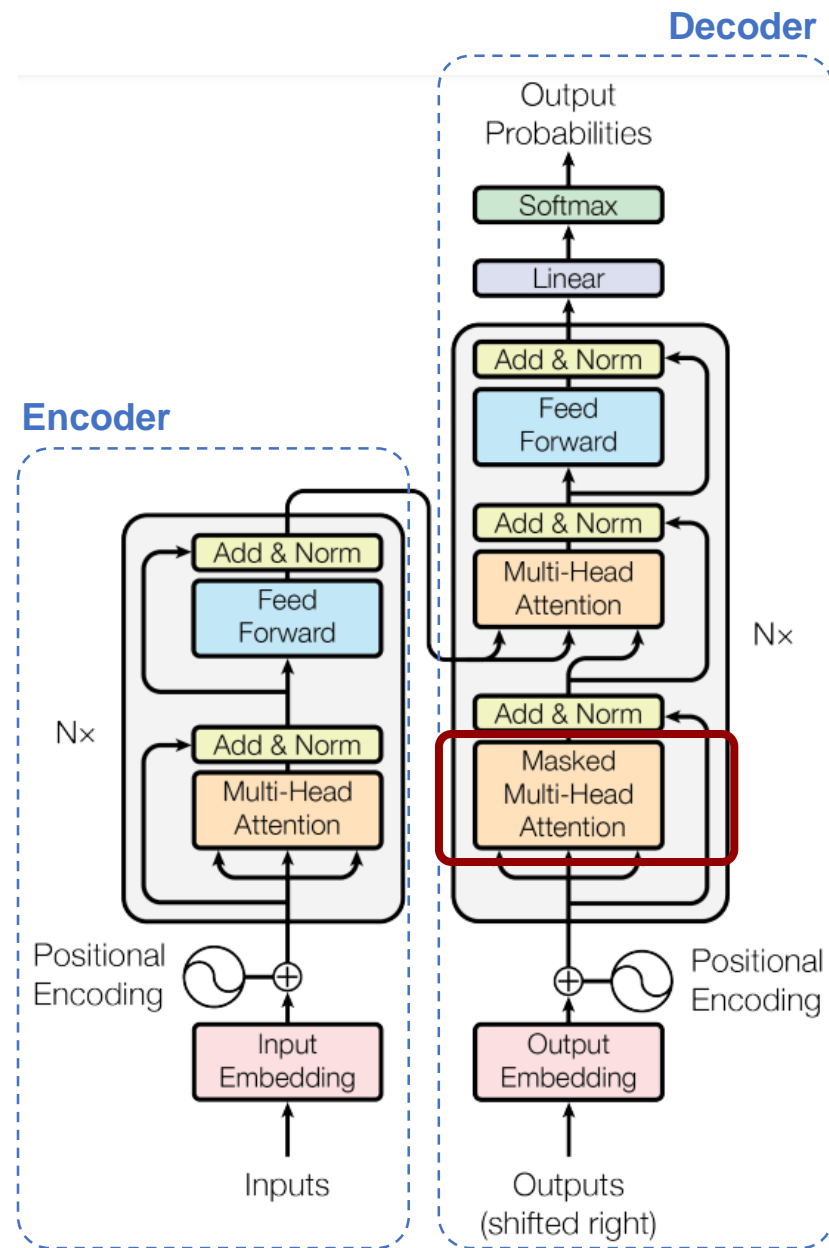


Figure 1: The Transformer - model architecture.

TTS 程式實作進度

目前已使用實際值確認無誤的部分

輸入已處理好的 input JSON 文檔

Transformer
(embed)

Embedding
ScaledPositionalEncoding x 6

(encoder)

(self_attention): Multi-Head Attention

(linear_q): **Linear()**

(linear_k): **Linear()**

(linear_v): **Linear()**

(linear_out): **Linear()**

(normalize): **LayerNorm()**

(feed forward):

(w_1): **Linear()**

(w_2): **Linear()**

ReLU()

(normalize): **LayerNorm()**

(embed)

(prenet)

Linear()

ReLU()

x 6

ScaledPositionalEncoding

(self_attention): Multi-Head Attention

(linear_q): **Linear()**

(linear_k): **Linear()**

(linear_v): **Linear()**

(linear_out): **Linear()**

(normalize): **LayerNorm()**

(src_attention): Multi-Head Attention

(linear_q): **Linear()**

(linear_k): **Linear()**

(linear_v): **Linear()**

(linear_out): **Linear()**

(normalize): **LayerNorm()**

(feed forward):

(w_1): **Linear()**

(w_2): **Linear()**

ReLU()

(normalize): **LayerNorm()**

x 4

(postnet)

Conv1d()

BatchNorm1d()

Tanh()

綠色：已完成底層功能，但仍需確認演算法，與運算結果

- Embedding
- ScaledPositionalEncoding
- Linear
- LayerNorm
- ReLU

紅色：尚未實作

- Conv1d()
- BatchNorm1d()
- Tanh()

時程表

- 尚未完成的部分
 - ❑ 整理Decoder演算法
 - ❑ 完成Decoder (所使用到的底層與Encoder相同)
 - ❑ postnet中的三種底層layer
 1. Conv1d()
 2. BatchNorm1d()
 3. Tanh()
 - ❑ 完成postnet
- 目前預估每一塊需一個禮拜的時間，因此預計需4個禮拜完成

