EMORY UNIVERSITY SCHOOL OF MEDICINE

A Tutorial for Transformer-based Classification Model

Presented by Yuting Guo
Email: yuting.guo@emory.edu

Outline

- Model Architecture
- Encoder Implementation
- Classification example code
- Practice tips

EMORY UNIVERSITY SCHOOL OF MEDICINE

Model Architecture

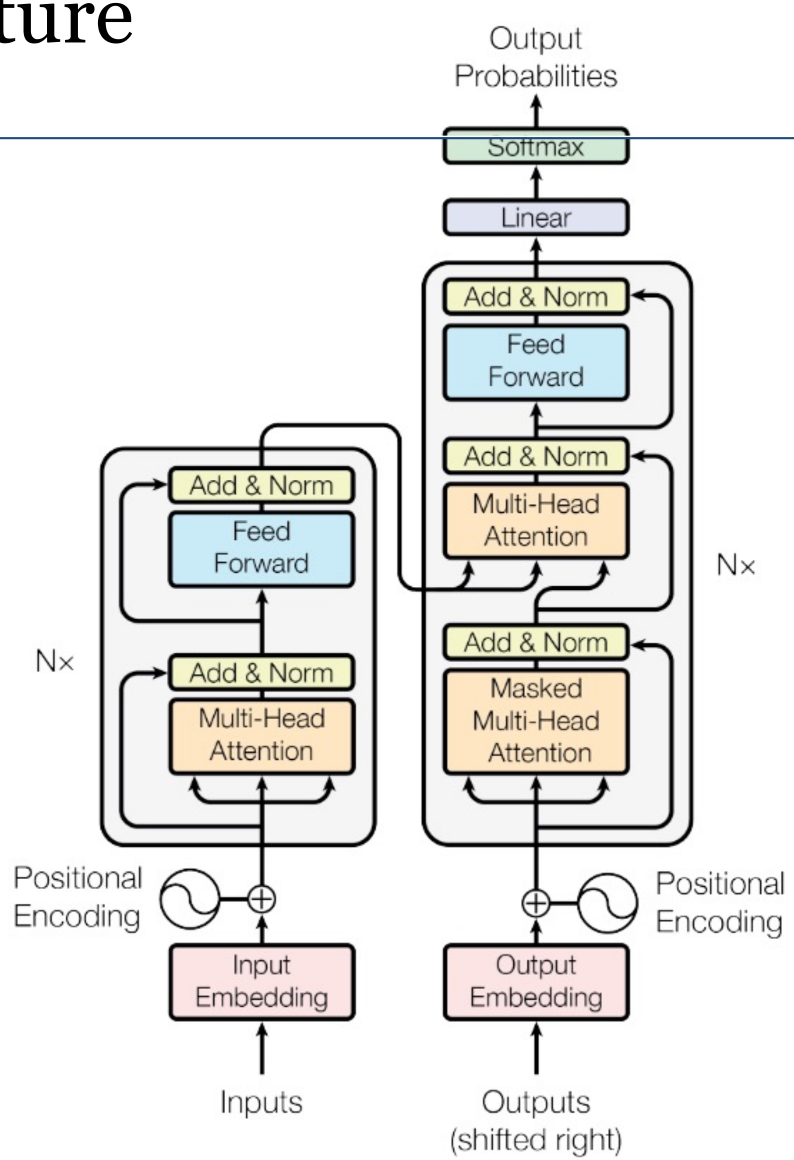


Figure 1: The Transformer model architecture.

Vaswani A, Shazeer N, Parmar N, et al. Attention is All you Need. In: Guyon I, Luxburg U Von, Bengio S, et al., eds. *Advances in Neural Information Processing Systems*. Vol 30. Curran Associates, Inc.; 2017:5999-6009.

EMORY UNIVERSITY SCHOOL OF MEDICINE

Model Architecture

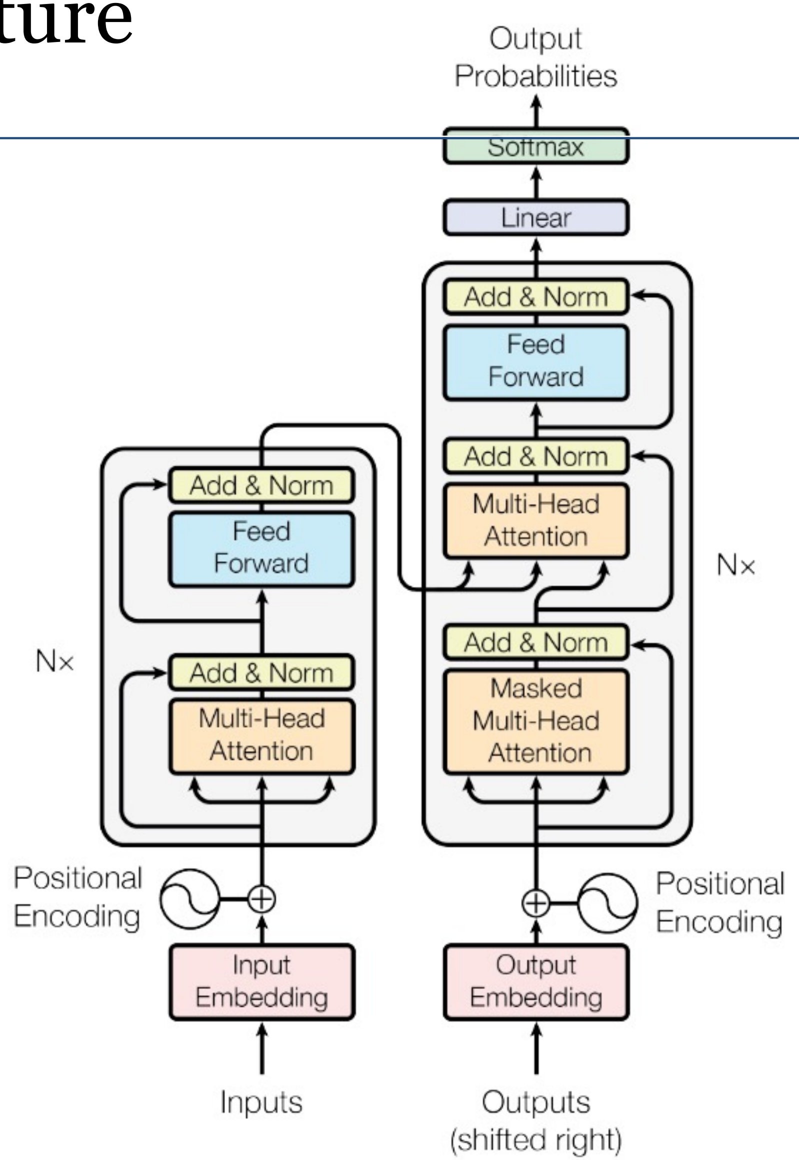
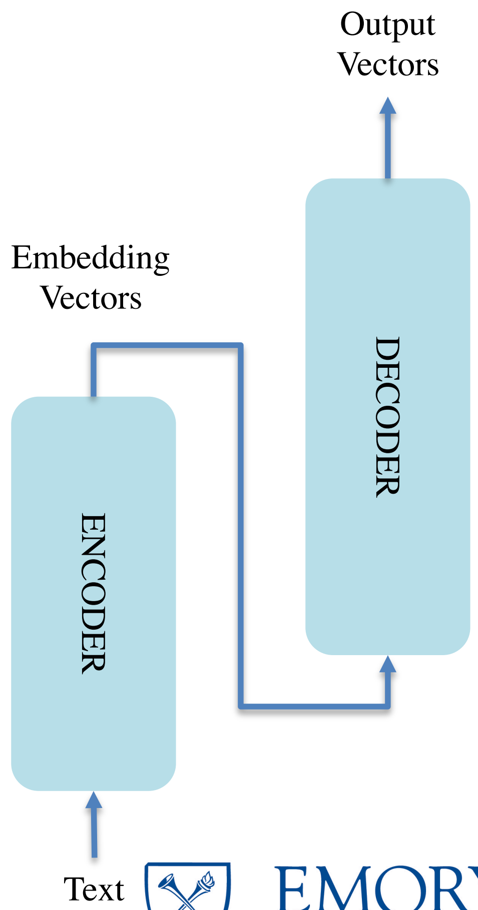


Figure 1: The Transformer model architecture.



Vaswani A, Shazeer N, Parmar N, et al. Attention is All you Need. In: Guyon I, Luxburg U Von, Bengio S, et al., eds. *Advances in Neural Information Processing Systems*. Vol 30. Curran Associates, Inc.; 2017:5999-6009.

EMORY UNIVERSITY SCHOOL OF MEDICINE

Model Architecture

➢ The classification model uses the encoder from a Transformer model.

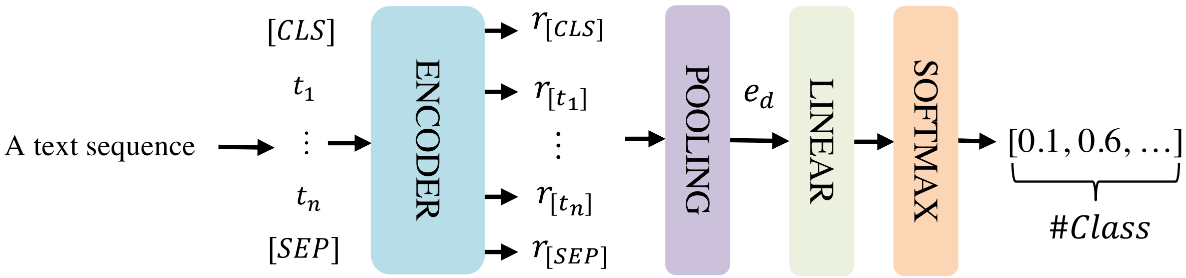


Figure 2: The Transformer-based classification model architecture.

Notations
[CLS]: a special token added to the beginning of the text.
[SEP]: a special token added to the end of the text.
$r_{[t]}$: the vector representation of a word or word piece t.
e_d: the vector representation of the text sequence.

EMORY UNIVERSITY SCHOOL OF MEDICINE

Encoder

- Key components/layers
  - input embedding
  - positional encoding
  - multi-head attention
  - layer normalization (add & norm)
  - feed forward

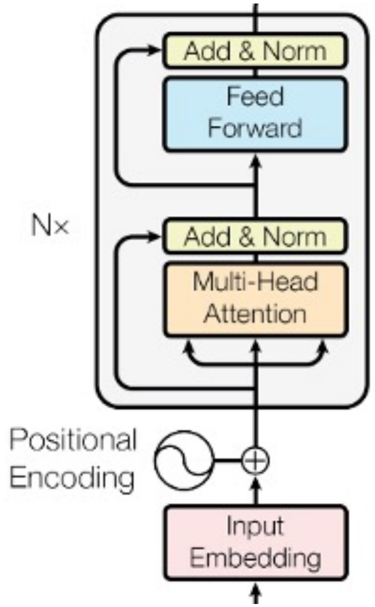


Figure 3: The Transformer encoder architecture.

Input Embedding

- Tokenization: text -> word pieces
- Vectorization: word pieces -> word embeddings

word embeddings	$v_{[I]}$ $v_{[have]}$ $v_{[a]}$ $v_{[new]}$ $v_{[GP]}$ $v_{[\#\#U]}$ $v_{[!]}$
Word pieces	'I', 'have', 'a', 'new', 'GP', '##U', '!'
Raw text	I have a new GPU!

Figure 4: Example for tokenization and vectorization process.

EMORY UNIVERSITY SCHOOL OF MEDICINE

Input Embedding

- Padding

$$\left[\begin{array}{ccccc} \begin{bmatrix}0.1\\0.3\\ \vdots\\0.6\end{bmatrix} & \begin{bmatrix}0.4\\0.2\\ \vdots\\0.3\end{bmatrix} & \cdots & \begin{bmatrix}0\\0\\ \vdots\\0\end{bmatrix} & \begin{bmatrix}0\\0\\ \vdots\\0\end{bmatrix}\end{array}\right]$$

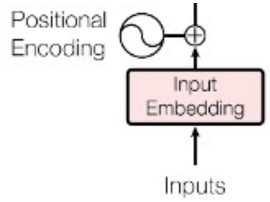
$v_{[I]}$ $v_{[have]}$ $v_{[<pad>]}$ $v_{[<pad>]}$

max_seq_len

Notations

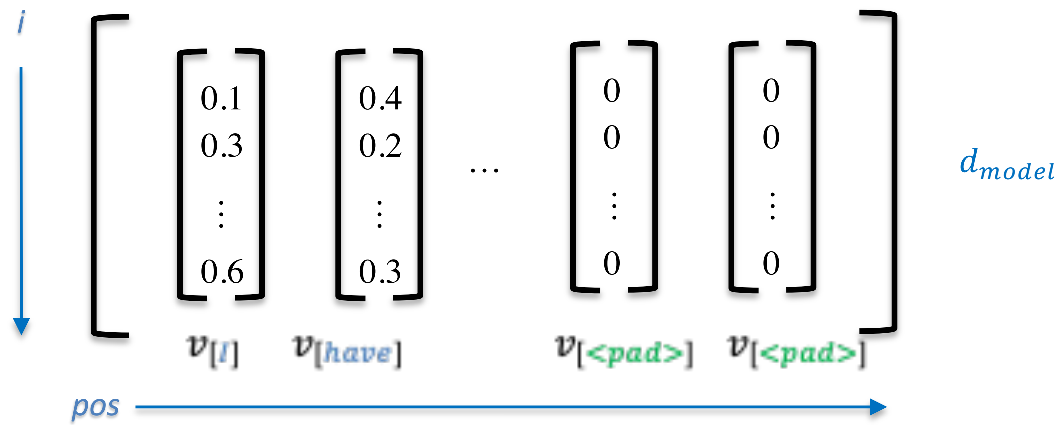
max_seq_len: the maximum sequence length.

Positional Encoding



- Generate a new matrix based on its position

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

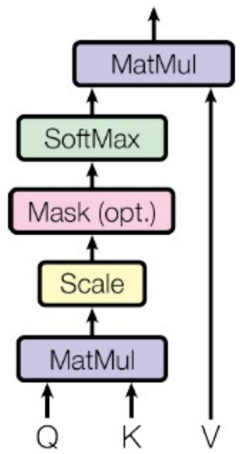


Notations
d_{model}: the word embedding size.
pos: the position of the word in the sequence
i: the dimension

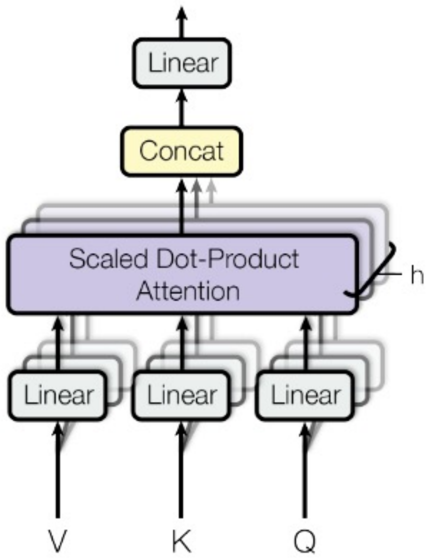
Multi-Head Attention

- Query (Q), Key (K), and Value (V)
- In BERT, Q = K = V ($d_{model} = d_k = d_v$)

Scaled Dot-Product Attention



Multi-Head Attention



$$\mathrm{Attention}(Q,K,V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

$$\mathrm{MultiHead}(Q,K,V) = \mathrm{Concat}(\mathrm{head_1},...,\mathrm{head_h})W^O$$
$$\text{where } \mathrm{head_i} = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

EMORY UNIVERSITY SCHOOL OF MEDICINE

Layer Normalization

- Normalization helps
  - reduce training time
  - unbiased model to higher value features
  - restrict weights to a certain range

$$x_{norm} = \frac{x - avg(x)}{\sqrt{var(x)}}$$

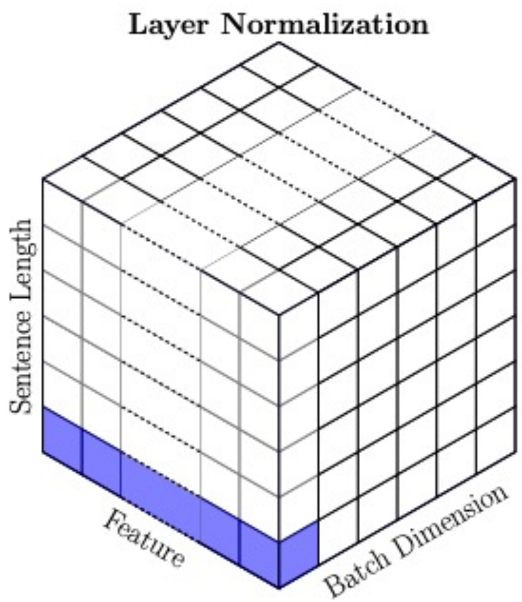


Figure 5: Layer normalization visualization.

https://www.kaggle.com/code/halflingwizard/how-does-layer-normalization-work

EMORY UNIVERSITY SCHOOL OF MEDICINE

Feed Forward

- Two linear transformations with a ReLU activation in between

$$\mathrm{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$$

Pooling

- As we know, the output of the encoder is a matrix of word emebddings. How can we convert that into a vector to present the entire sentence/document?
  - Averaging pooling
  - Max pooling
  - Use the special token [CLS] or [SEP]

EMORY UNIVERSITY SCHOOL OF MEDICINE

Open Source Library

- Transformers (HuggingFace)
  - https://github.com/huggingface/transformers
- SimpleTransformers
  - https://github.com/ThilinaRajapakse/simpletransformers
- https://github.com/CyberZHG/torch-multi-head-attention

EMORY UNIVERSITY SCHOOL OF MEDICINE

Classification Example Code

- Data process
- Model training
- Model inference

EMORY UNIVERSITY SCHOOL OF MEDICINE

Practice Tips

- Important hype-parameters for model performance
  - Maximum sequence length
  - Batch size
  - Training time
  - Learning rate

EMORY UNIVERSITY SCHOOL OF MEDICINE

Practice Tips

- How to select a pre-trained Transformer-based model for your task?
  - Model's pre-training data
  - Model size (number of parameters)
    - BERT-large vs BERT-base
    - RoBERTa-large vs RoBERTa-base

Guo Y, Dong X, Al-Garadi MA, Sarker A, Paris C, Mollá-Aliod D. Benchmarking of Transformer-Based Pre-Trained Models on Social Media Text Classification Datasets. In: *Proceedings of the The 18th Annual Workshop of the Australasian Language Technology Association*. ; 2020:86-91.

EMORY UNIVERSITY SCHOOL OF MEDICINE