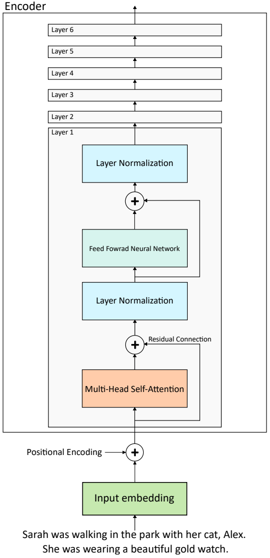
Transformers (Part 5)

Dr. Alireza Aghamohammadi

Transformer Encoder Architecture

- ❖ Transformer uses an encoder-decoder architecture.
- ❖ This lecture will focus on the encoder component.
- ❖ The encoder is composed of six layers. Each layer includes:
  - ❑ Multi-head self-attention mechanism
  - ❑ Layer normalization
  - ❑ Feed-forward neural network
- ❖ Each layer incorporates two residual connections to retain the input information.



Multi-Head Self Attention

- Instead of using a single set of weights for queries (W_q), keys (W_k), and values (W_v) with dimensions $\mathbb{R}^{d \times n}$, we use S different sets for our input $X \in \mathbb{R}^{n \times m}$.
- For queries, we have $W_q^1, W_q^2, \ldots, W_q^S \in \mathbb{R}^{\frac{d}{S} \times n}$, resulting in $Q_1, Q_2, \ldots, Q_S \in \mathbb{R}^{\frac{d}{S} \times m}$.
- For keys, we have $W_k^1, W_k^2, \ldots, W_k^S \in \mathbb{R}^{\frac{d}{S} \times n}$, resulting in $K_1, K_2, \ldots, K_S \in \mathbb{R}^{\frac{d}{S} \times m}$.
- For values, we have $W_v^1, W_v^2, \ldots, W_v^S \in \mathbb{R}^{\frac{d}{S} \times n}$, resulting in $V_1, V_2, \ldots, V_S \in \mathbb{R}^{\frac{d}{S} \times m}$.
- The attention matrix for the h^{th} head is computed as:

$$A_h = \texttt{softmax}\left(\frac{K_h^T \cdot Q_h}{\sqrt{d}}\right)$$

- The final hidden representation for the H heads are:

$$[V_1 A_1, V_2 A_2, \ldots, V_S A_S]^T \cdot W_o \in \mathbb{R}^{d \times m}$$

- The Transformer paper sets $d, n = 512$ and the number of heads $S = 8$.

Layer Normalization

- Layer normalization is a technique used to normalize the inputs of each layer in a neural network.
- **C**onsider the i^{th} layer of a neural network with D neurons: $h_{i1}, h_{i2}, \ldots, h_{iD}$.
- **Step 1:** Calculate the mean of the layer's output:

$$\mu_L = \frac{1}{D} \sum_{j=1}^{D} h_{ij}$$

- **Step 2:** Calculate the variance of the layer:

$$\sigma_L^2 = \frac{1}{D} \sum_{j=1}^{D} (h_{ij} - \mu_L)^2$$

- **Step 3:** Normalize the input, where ϵ is a small constant to avoid division by zero:

$$\hat{h}_{ij} = \frac{h_{ij} - \mu_L}{\sqrt{\sigma_L^2 + \epsilon}}$$

- **Step 4:** Scale and shift, where γ and β are learnable parameters:

$$y_{ij} = \gamma \hat{h}_{ij} + \beta$$

Feed-Forward Blocks in Transformer

- Each encoder layer in a transformer includes a feed-forward network block, which consists of two linear layers with a ReLU activation function in between.
- The feed-forward network (FFN) can be represented as:

$$\mathsf{FFN}(x) = W_2^T \cdot \mathsf{ReLU}(W_1^T \cdot x + b_1) + b_2$$

- Here, $X \in \mathbb{R}^{512\times m}$ is the input, $W_1 \in \mathbb{R}^{512\times 2048}$ and $W_2 \in \mathbb{R}^{2048\times 512}$ are weight matrices, $b_1 \in \mathbb{R}^{2048}$ and $b_2 \in \mathbb{R}^{512}$ are bias vectors.
- The input and output of the feed-forward block have the same dimensionality.