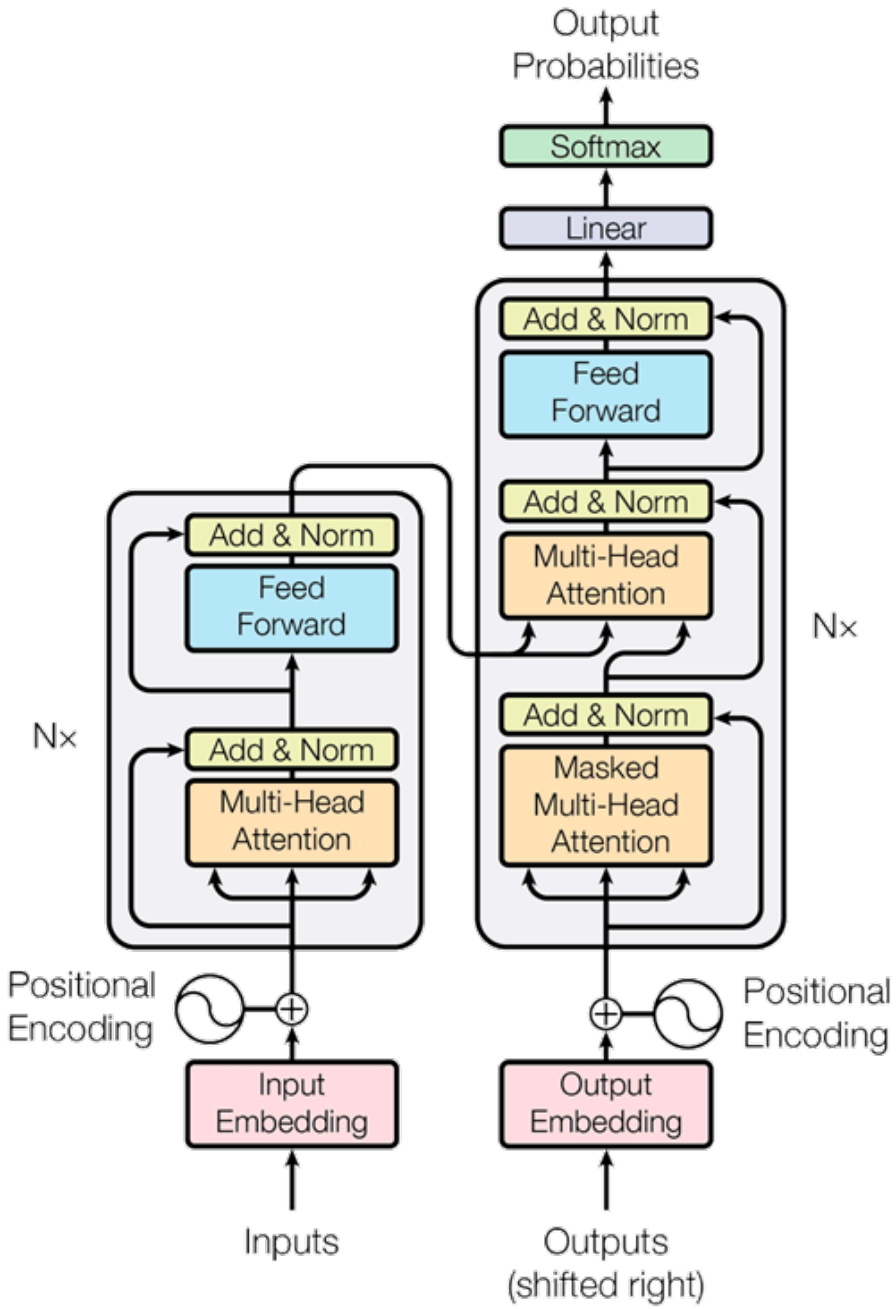
# Transformer: Encoder Block

Ki Hyun Kim

nlp.with.deep.learning@gmail.com

# Transformer



# Equations

- Given Dataset,

$$\mathcal{D} = \{x^i, y^i\}_{i=1}^N$$
$$x^i = \{x_1^i, \cdots, x_m^i\} \text{ and } y^i = \{y_0^i, y_1^i, \cdots, y_n^i\},$$
$$\text{where } y_0 = <\text{BOS}> \text{ and } y_n = <\text{EOS}>.$$

- What we want is

$$\hat{y}_{1:n} = f(x_{1:m} : \theta)$$

# Equations

- $Q$, $K$ and $V$ are from previous layer:
  - Residual connections and Layer Normalizations are used.

$$h_{0,1:m} = \mathrm{emb}(x_{1:m}) + \mathrm{pos}(1, m)$$

$$\tilde{h}^{\mathrm{enc}}_{i,1:m} = \mathrm{LayerNorm}(\mathrm{Multihead}_i(Q, K, V) + h^{\mathrm{enc}}_{i-1,1:m}),$$

$$\text{where } Q = K = V = h^{\mathrm{enc}}_{i-1,1:m}.$$

$$\mathrm{FFN}(h_{i,t}) = \mathrm{ReLU}(h_{i,t} \cdot W^1_i) \cdot W^2_i$$

$$\text{where } W^1_i \in \mathbb{R}^{d_{\mathrm{model}} \times d_{\mathrm{ff}}} \text{ and } W^2_i \in \mathbb{R}^{d_{\mathrm{ff}} \times d_{\mathrm{model}}}.$$

$$h^{\mathrm{enc}}_{i,1:m} = \mathrm{LayerNorm}([\mathrm{FFN}(\tilde{h}^{\mathrm{enc}}_{i,1}); \cdots; \mathrm{FFN}(\tilde{h}^{\mathrm{enc}}_{i,m})] + \tilde{h}_{i,1:m})$$

# Equations

- Encoder is stack of encoder blocks:

$$h^{\text{enc}}_{\ell_{\text{enc}},1:m} = \text{Block}_{\text{enc}}(h^{\text{enc}}_{\ell_{\text{enc}}-1,1:m})$$
$$\dots$$
$$h^{\text{enc}}_{1,1:m} = \text{Block}_{\text{enc}}(h^{\text{enc}}_{0,1:m})$$

# Summary

- Encoder는 self-attention으로 구성되어 있음
  - $Q, K, V$는 이전 레이어의 출력 값 – 즉, 같은 값

- Seq2seq의 attention과 달리, $Q$도 모든 time-step을 동시에 연산
  - 빠르지만 굉장히 메모리를 많이 먹게 됨

- Residual connection으로 인해 깊은 네트워크 구성 가능
  - Big LM의 토대 마련