**Transformer**

**Self-attention: multi-headed**

- Residual
- Layer Norm
- FNN (element-wise)
  - Input: \( X \)
  - Output:
    - \( W_i \) for \( i = 1 \ldots 9 \) heads
    - \( d_k = 64 \)
    - \( d_q = d_k \)
    - \( d_v = 64 \)
    - \( V'_i = \text{Softmax} \left( \frac{Q_i K_i^T}{\sqrt{d_k}} \right) \cdot V_i \)
    - Concat \( V'_i \to V' \) (\( h \cdot d_v \))
    - Weight matrix \( W^o \)

**Layer Norm**

- Input: \( x_1 \)
- Output: \( x'_1 \)
- After SA

**Scaler**

- Input: \( x \)
- Output: \( \mu \), \( \sigma \)
Transformer block

$d_{model} = 512$

Number of layers of transformer blocks

Final output embeddings

Pretraining

Word2vec: 1 token/word → 1 embedding

9 was sitting by the river bank →

9 went to the bank and sat on the bench →

Transformer:

1 token/word → multiple embeddings (as many blocks/occurrences)
Pos embedding: 1) Cos/sin based
2) preserve relative pos info
Input block

\[ \begin{bmatrix} x_1 & x_2 \end{bmatrix} \to \begin{bmatrix} x_N \end{bmatrix} \]

512 dim

for \( n = 1 \) to \( N \)

\( p_{r, 2^j n} \) = \sin \left( \frac{r_{os} (10000)^{2^j/d_{w1}}} {\text{dim}} \right) \)

\( p_{w, 2^j n} \) = \cos \left( \frac{r_{os} (10000)^{2^j/d_{w1}}} {\text{dim}} \right) \)

Continuous version of binary encoding

0 → 0000
1 → 0001
2 → 0010
3 → 0011
4 → 0100
**BERT**: pre-trained model to generate contextual embeddings

- bare model
  - $d_{\text{model}} = 768$
  - block size = 512 = $n$  \( x_1, \ldots, x_n \)
  - \# of layers = 12
    - (transformer layers)
  - \# of heads = \( q/12 \) ?
    - \( d_k = d_v = d_{q=6} \)

30k vocab of tokens
(word-piece)

MLM: Masked Language Model
(self-supervised)

The quick brown fox jumped -

N = 572
I) MLM ← loss CE over 30k vocab per masked position

II) NSP ← next sentence prediction

\[
\begin{align*}
\text{NSP} \quad &\quad S_1 | \text{sep} | S_2 \quad \rightarrow \quad 1 \\
\text{NSP} \quad &\quad S_1 | \text{sep} | S_2 \quad \rightarrow \quad 0
\end{align*}
\]

\[
L = \text{MLM loss} + \text{NSP loss}
\]

\[
X_i + p_i + e_{A/\theta}\]

\[
\text{BERT}
\]

\[
\text{segment embeddings}
\]

\[
2 \times 768
\]

\[
\text{CPLN}
\]

\[
\text{binary CE}
\]
\[ T = \text{nn. Embedding}(V, d) \]

\[ T(100) \rightarrow C([5, 6, 3, 4]) \]

\[ \text{logits} = \text{torch.sum}(T' \times C', \text{dim}=1) \]
\[ \text{Np. search sorted } (\ldots) \]
\[ (0.1, 0.5, \ldots) \]
\[ \text{Np. random choice } \left( \frac{\text{vec1, vec2, vec3}}{1000} \right) \]
\[ \text{Randen } = 0.1 \quad \text{Pxx 2} \]
\[ 0.5 \quad \text{Pxx 1810} \]