Input.

Q) T → 2 tokens → "cat sat"

(V) Vocab size (3) → IDs → 0: the, 1: cat, 2: sat

Embedding size (channels (V)) → 2

(h) Attention heads: 1 → $d_k = C/h = 2$

cat sat

$x = [1, 2]$ cat sat

Weights] Token Embedding table $W_{te} \in R^{V \times C}$

$$W_{te} = \begin{bmatrix} 0.1 & 0.2 \\ 1.0 & 0.5 \\ 0 & 2.0 \end{bmatrix} \rightarrow \text{Vocab.} \quad \begin{matrix} \text{the} \\ \text{cat} \\ \text{sat} \end{matrix}$$

↳ 2 features

- Positional Embedding table $W_{pe} \in R^{T \times C}$

$$W_{pe} = \begin{bmatrix} 0.5 & -0.5 \\ 1.0 & 1.0 \end{bmatrix} \begin{matrix} x \\ x[0] \\ x[1] \end{matrix} \quad \text{sequence inputs.}$$

step 1] lookup and addition. $X = W_{te} + W_{pe}$

in W_{te} we check on token id.

so $x[0]$ = cat → token id = 1.

∴ we select $W_{te}[1]$ row

doing with mat $X = \begin{bmatrix} 1.0 & 0.5 \\ 0 & 2.0 \end{bmatrix} + \begin{bmatrix} 0.5 & -0.5 \\ 1.0 & 1.0 \end{bmatrix}$

Combined input mat $X = \begin{bmatrix} 1.5 & 0 \\ 1.0 & 3 \end{bmatrix}^T$

↳ C

Step 2] Pre-layer Norm.
normalize each row (token) across channels using
$\gamma = [1.0, 1.0]$ and $\beta = [0.0, 0.0]$

Token 0 $x_0 = [1.5, 0]$
mean(μ_0) = $1.5/2 = 0.75$

Variance (σ_0^2) $= \frac{1}{C} \sum_{i=1}^{C=2} (x_i - \mu_0)^2$

$= \frac{(1.5 - 0.75)^2 + (0 - 0.75)^2}{2}$

$\sigma_0 = 0.75$

Standardize = $\frac{x_{00} - \mu_0}{\sigma_0 + \varepsilon} = \left[\frac{1.5 - 0.75}{0.75}, \frac{0 - 0.75}{0.75}\right]$ (x_{00}, x_{01})
[1 x C]

$= [1.0, -1.0]$

Scale and shift $\Rightarrow Y_0 = x_0 \cdot \gamma + \beta \rightarrow [1.0, -1.0]$.

Token 1 $= [1., 3]$; $(\mu_1) = 2$.
$\sigma_1^2 = 1 \neq 1 \rightarrow \sigma_1 = 1$.
$Y_1 = [-1, 1]$

$x_{norm} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$ ← Normalized input matrix.

$x_{norm} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$

Step 3] Compute Q, K, V projections.

$W_q = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$, $W_K = \begin{bmatrix} 0.5 & 1 \\ 1 & 0.5 \end{bmatrix}$, $W_V = \begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix}$

$$\begin{bmatrix} Q \\ K \\ V \end{bmatrix} = \begin{bmatrix} x_{norm} \end{bmatrix} \cdot \begin{bmatrix} W_Q \\ W_K \\ W_V \end{bmatrix}$$

$Q = \begin{bmatrix} 1 & -1 \\ -2 & 2 \end{bmatrix}$, $K = \begin{bmatrix} -0.5 & 0.5 \\ 0.5 & -0.5 \end{bmatrix}$, $V = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$

$Q = \begin{bmatrix} 1 & -2 \\ -1 & 2 \end{bmatrix}$ $K = \begin{bmatrix} -0.5 & 0.5 \\ 0.5 & -0.5 \end{bmatrix}$ $V = \begin{bmatrix} 2 & 0 \\ -2 & 0 \end{bmatrix}$

Step 4] Casual self attention. Score $= \frac{Q \cdot K^T}{\sqrt{d_K}}$

$K^T = \begin{bmatrix} -0.5 & 0.5 \\ 0.5 & -0.5 \end{bmatrix}$; $\sqrt{d_K} = \sqrt{2} = 1.414$

$$\text{Attention Scores}(A) = \begin{bmatrix} \frac{-1.5}{\sqrt{2}} & \frac{1.5}{\sqrt{2}} \\ 1.5/\sqrt{2} & \frac{-1.5}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} -1.06 & 1.06 \\ 1.06 & -1.06 \end{bmatrix}$$

Apply mask on upper triangle

$$A_{masked} = \begin{bmatrix} -1.06 & -\infty \\ 1.06 & -1.06 \end{bmatrix}$$ where $j > i \rightarrow -\infty$.

Step 5] Row wise softmax $(x_i) = \frac{e^{x_i}}{\sum_{k=1}^{K} e^{x_k}}$ basically finding probabilities.

Row 0 -> -1.06

$\frac{e^{-1.06}}{e^{-1.06}} = 1$; $\frac{e^{-\infty}}{e^{-1.06} + \cdots} = \frac{0}{1} = 0.$

Row 1 -> $\frac{e^{1.06}}{e^{1.06} + e^{-\infty}} = 0.8928$; $\frac{e^{-\infty}}{\cdots} = 0.1072.$

Attention weights $(A) = \begin{bmatrix} 1.0000 & 0.0000 \\ 0.8928 & 0.1072 \end{bmatrix}$

multiply by val matrix.

$$O = A \cdot V = \begin{bmatrix} 1 & 0 \\ 0.8928 & 0.1072 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ -2 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 2 & 0 \\ 1.5712 & 0 \end{bmatrix}.$$

Step 6] Add Residual 1. $x_{res1} = O + x$.

To preserve og. feature mag.

$$x_{res1} = \begin{bmatrix} 1.5 & 0 \\ 1 & 3 \end{bmatrix} + \begin{bmatrix} 2 & 0 \\ 1.5172 & 0 \end{bmatrix} = \begin{bmatrix} 3.5 & 0 \\ 2.5172 & 3 \end{bmatrix}$$

Step 7] Pre LN 2 giving us

$$x_{norm2} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

As seen all our mag is lost. cause $C = 2$, LN acts as a mere sign operator.

eg -> $\hat{x}_1 = \frac{a - \frac{a+b}{2}}{\left|\frac{a-b}{2}\right|}$ -> mean / -> sd $= 1.$

$C =$

$C = 2$ trap. models like GPT2 use 768 LLam use $C = 4096$.

FFN.

Step 8) Feed forward net / Multilayer Perceptron (MLP)
FFN expands channels from $C \to 4C$.
applies GELU (earlier used to apply RELU)
and projects back from $4C \to C$

$FFN(x) = Gelu(x \cdot W_1) \cdot W_2$.
applied on every single token.

a) first $x \cdot W_1 = \begin{bmatrix} 1.0 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \\ 0 & 2 & 1 \end{bmatrix}$

$= \begin{bmatrix} 1 & -2 & -2 \\ -1 & 2 & 2 \end{bmatrix}$

b) Apply GELU → Gaussian Error Linear Unit activation

$GELU(x) = x \cdot \Phi(x)$ → Cumulative dist. func. of std norm dist.

$\therefore GELU(x) = 0.5\,x \left[1 + \tanh\left(\sqrt{\frac{2}{\pi}}(x + 0.044715\,x^3)\right)\right]$

Giving output = $\begin{bmatrix} 0.8413 & -0.0454 & -0.0454 \\ -0.1587 & 1.9547 & 1.9547 \end{bmatrix}$

c) Projecting back to $C = 2$.

$FFN_{out} = H_{gelu} \cdot W_2$.

$FFN_{out} = \begin{bmatrix} 0.7959 & -0.0908 \\ 1.7960 & 3.9094 \end{bmatrix}$

Step 9] Add Residual 2. → $x_{out} = x_{res_1} + FFN_{out}$

$$x_{out} = \begin{bmatrix} 4.2959 & -0.0908 \\ 4.3672 & 6.9094 \end{bmatrix}$$

final sub layer of transformer block.

Step 10] finally project x_{out} to language model (LM) head to calc logits over our vocab (V=3)

W_{LM} → weights of dim of token embedding W_{te}. but to match x_{out}, instead of $V \times C$ we set W_{LM} as $C \times V$.

$$\therefore W_{LM} = \begin{bmatrix} 1 & 0 & 0.5 \\ 0 & 1 & 0.5 \end{bmatrix}$$

$$\text{logits} = x_{out} \cdot W_{LM}.$$

$$= \begin{bmatrix} 4.2959 & -0.0908 \\ 4.3672 & 6.9094 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0.5 \\ 0 & 1 & 0.5 \end{bmatrix}$$

$2 \times 2 \qquad 2 \times 3.$

$$\text{Logits} = \begin{bmatrix} 4.2959 & -0.0908 & 2.1025 \\ 4.3672 & 6.9094 & 5.6383 \end{bmatrix}$$

Calc softmax. $\frac{e^{x_i}}{\sum e^{x_k}}$ for Logits matrix. Input ↓

Full softmax mat (probabilities) $= \begin{bmatrix} 0.8897 & 0.0111 & 0.0992 \\ 0.0579 & 0.7357 & 0.2064 \end{bmatrix}$ the ← cat, sat ← cat, sat

So when it sees cat → the w 88% prob but ignores cat, sat → cat w 73% prob.