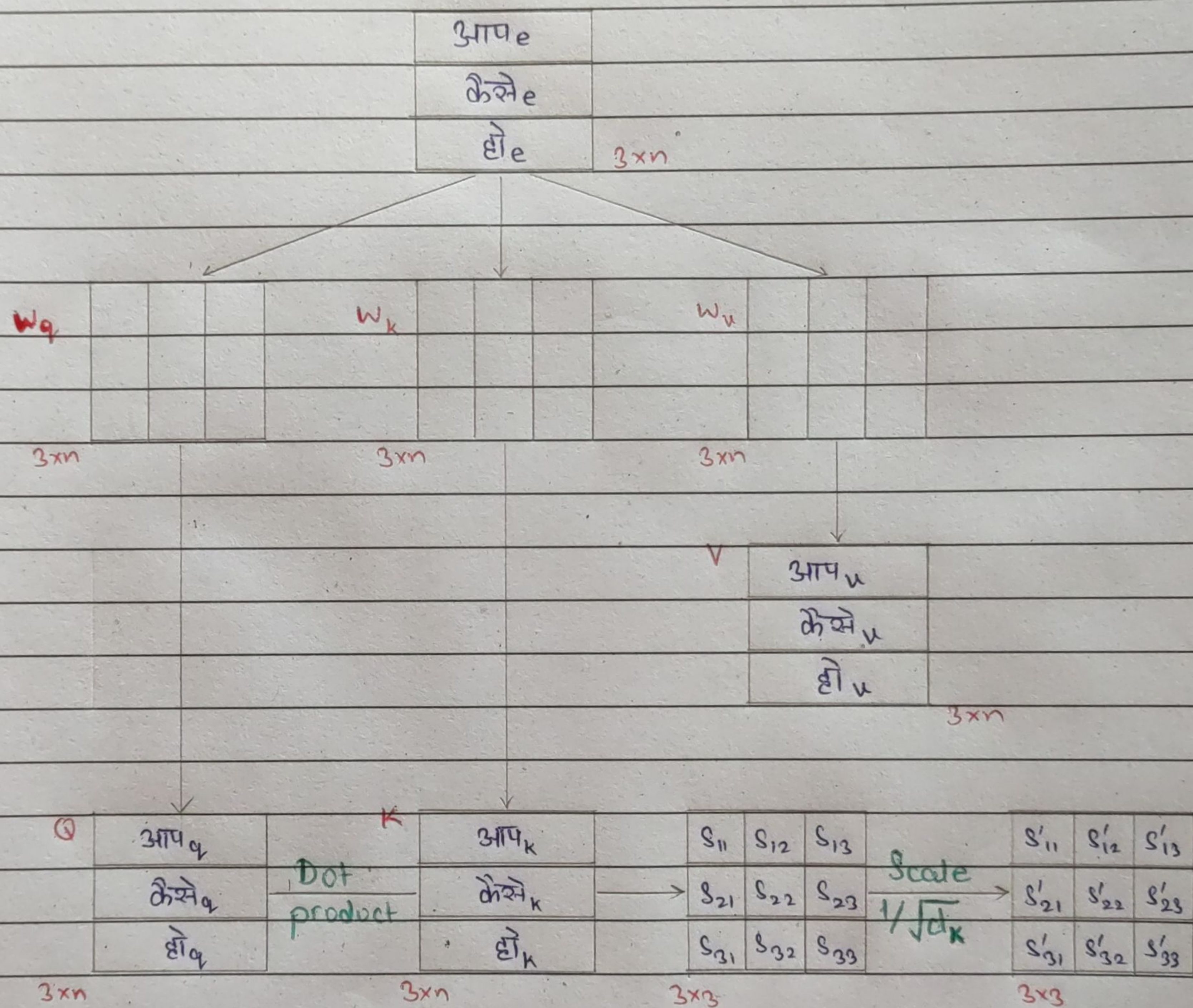
Consider the input sequence to be "How are you" with output sequence to be "आप कैसे हो". Let the embeddings of output sequence with positional encoding be $आप_e$, $कैसे_e$, $हो_e$.



* d_k : Dimension of key vectors in K-Matrix.

When using normal self-attention:

$$\begin{bmatrix} S'_{11} & S'_{12} & S'_{13} \\ S'_{21} & S'_{22} & S'_{23} \\ S'_{31} & S'_{32} & S'_{33} \end{bmatrix} \xrightarrow{\text{Softmax}} \begin{bmatrix} W_{11} & W_{12} & W_{13} \\ W_{21} & W_{22} & W_{23} \\ W_{31} & W_{32} & W_{33} \end{bmatrix}$$

$$\Rightarrow आप_{ce} = w_{11} \times आप_v + w_{12} \times कैसे_v + w_{13} \times हो_v$$

$$कैसे_{ce} = w_{21} \times आप_v + w_{22} \times कैसे_v + w_{23} \times हो_v$$

$$हो_{ce} = w_{31} \times आप_v + w_{32} \times कैसे_v + w_{33} \times हो_v$$

Now, as we can observe, in order to avoid the problem of data leakage in non-autoregressive training of the decoder we need to prevent the contribution of कैसे$_u$ & हो$_u$ in calculating contextual embedding for आप$_{ce}$. |||ly, we need to prevent contribution of हो$_u$ in calculating contextual embedding for कैसे$_{ce}$.

This is because in normal self-attention mechanism the calculation of आप$_{ce}$ is relying on कैसे$_u$ & हो$_u$ along with आप$_u$. This cannot happen as these are future values & decoder won't have access to them at inference time. This will create the problem of data leakage. Similar is the case when कैसे$_{ce}$ where हो$_u$ is the future value कैसे$_{ce}$ is relying on that it can't have access to at inference. But it can have access to आप$_u$ for calculating as it has already come into picture before it.

Therefore, in relevance to this example, we need to prevent the contribution of कैसे$_u$, हो$_u$ from आप$_{ce}$ eq^n & हो$_u$ from कैसे$_{ce}$ eq^n. This can be done by turning their corresponding weights to 0, i.e. $w_{12} = w_{13} = w_{23} = 0$. This can be done using a mask matrix.

$$\underbrace{\begin{bmatrix} s'_{11} & s'_{12} & s'_{13} \\ s'_{21} & s'_{22} & s'_{23} \\ s'_{31} & s'_{32} & s'_{33} \end{bmatrix}}_{S'} + \underbrace{\begin{bmatrix} 0 & -\infty & -\infty \\ 0 & 0 & -\infty \\ 0 & 0 & 0 \end{bmatrix}}_{\text{Mask}} \longrightarrow \begin{bmatrix} s'_{11} & -\infty & -\infty \\ s'_{21} & s'_{22} & -\infty \\ s'_{31} & s'_{32} & s'_{33} \end{bmatrix} \xrightarrow{\text{Softmax}} \begin{bmatrix} w_{11} & 0 & 0 \\ w_{21} & w_{22} & 0 \\ w_{31} & w_{32} & w_{33} \end{bmatrix}$$

In general terms, a mask matrix of same dimensions as the S' matrix with upper diagonal values as $-\infty$ & rest as 0 is added to the S' matrix giving an altered S' matrix with upper diagonal values as $-\infty$ & rest as previous S' values.

After applying softmax on this altered S' matrix we get an altered weight matrix W where upper diagonal values are 0 & rest are regular outcomes of the softmax.

* softmax$(-\infty) = 0$

Due to this our eq^{n}s become as follows:

$$\text{आप}_{ce} = w_{11} \times \text{आप}_v + \cancelto{0}{0 \times \text{कैसे}_v} + \cancelto{0}{0 \times \text{हो}_v}$$
$$\text{कैसे}_{ce} = w_{21} \times \text{आप}_v + w_{22} \times \text{कैसे}_v + \cancelto{0}{0 \times \text{हो}_v}$$
$$\text{हो}_{ce} = w_{31} \times \text{आप}_v + w_{32} \times \text{कैसे}_v + w_{33} \times \text{हो}_v$$

~~Thus~~ $\therefore$, we have successfully prevented use of future value embeddings in calculating contextual embeddings of certain words, thus avoiding the problem of data leakage while also keeping non-autoregressive training of decoder.