Input Transformation In Multi-Headed Attention.

Symbols:

d_model → Input vector size (Embedding size for first layer)

d_k → Length(Size) of the Query, key and Value vectors in each head.

n_h → Number of attention heads.

n_t → Number of tokens in a sentence length.

t_{ij} → Vector for token (j) in sentence (i)

q_{ijk} → Query vector for token (j) in sentence (i) and head (k).

n_b → batch size (Number of sentences in a batch)

k_{ijk} → key vector for token (j) in sentence (i) and head (k).

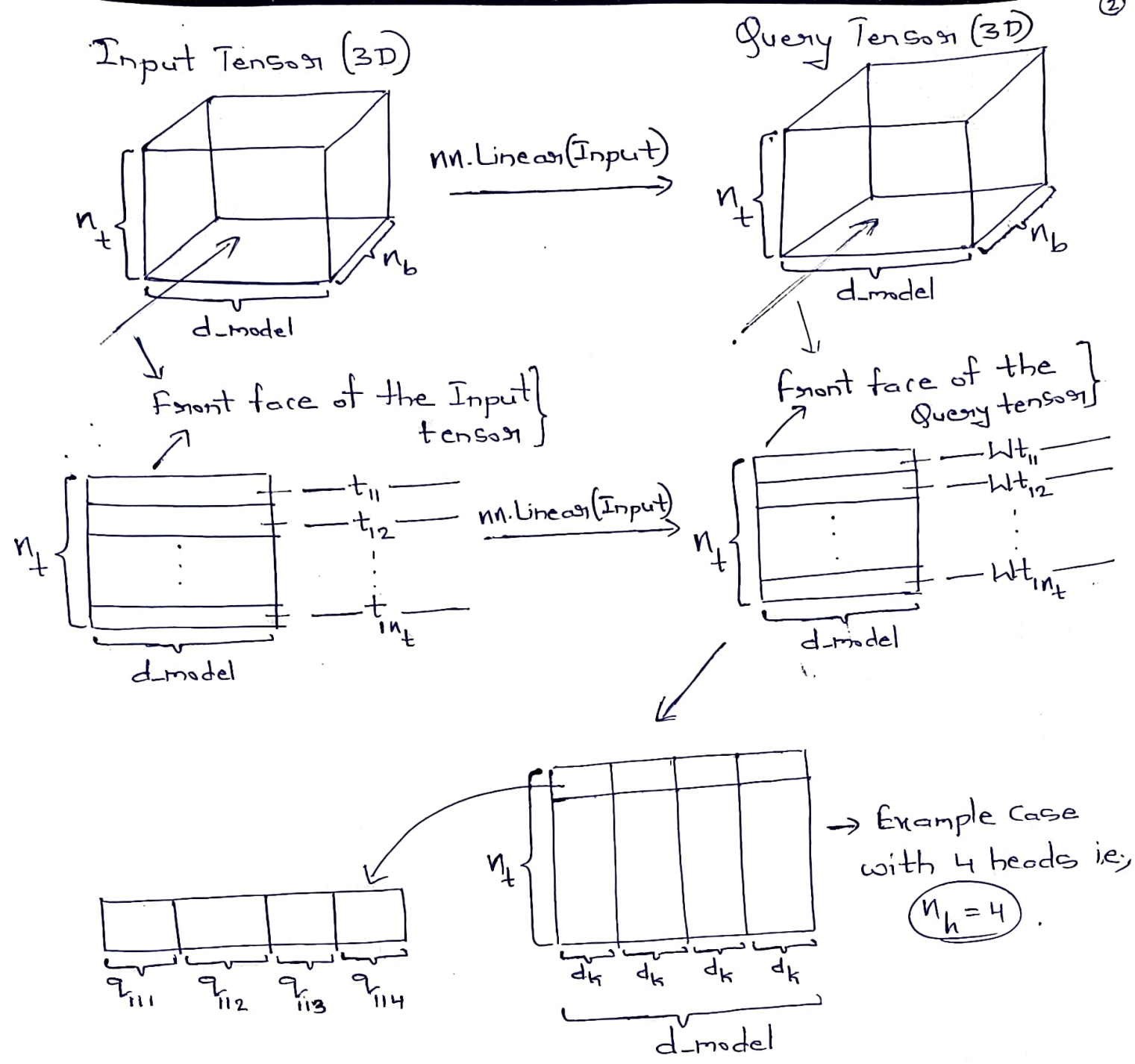
v_{ijk} → Value vector for token (j) in sentence (i) and head (k).

a_{ijk} → Attention score vector for token (j) in sentence (i) and head (k). {from all other tokens in the sentence i.}

ah_{ijk} → Attention head for token (j) in sentence (i) and head (k).

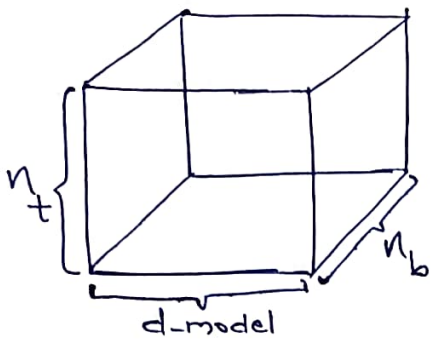
c_{ij} → Concatenated Attention vector for token (j) in sentence (i).

m_{ij} → Output vector for token (j) in sentence (i) from the Multi-Headed attention layer.



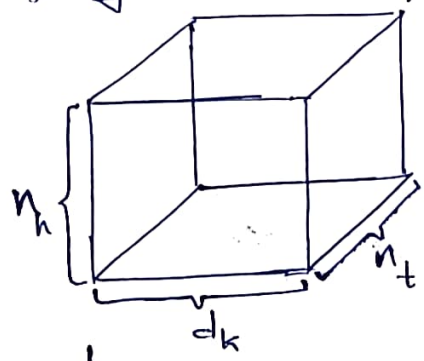
{We are calculating query vectors for all the heads using a single matrix multiplication. (or a single nn.Linear layer)}

Query Tensor (3D)



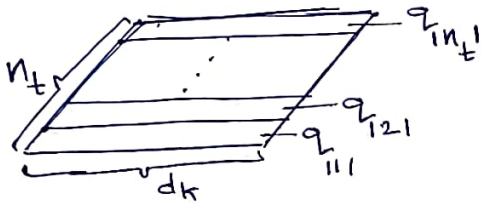
tensor.view()

Query Tensor Reshaped (4D)

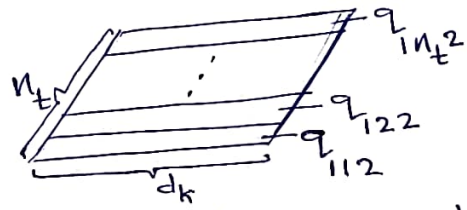


n_b → Number of these 3D tensors in the 4D Query tensor.

{Omitted batch dimension since I cannot draw a 4D tensor}

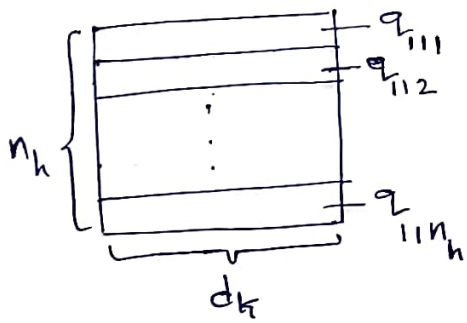


Top view of the first layer in the 4D Query tensor.

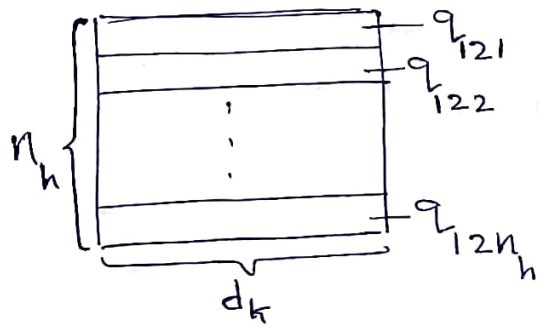


Top view of the second layer in the 4D Query tensor.

There will be n_h layers from Top view.



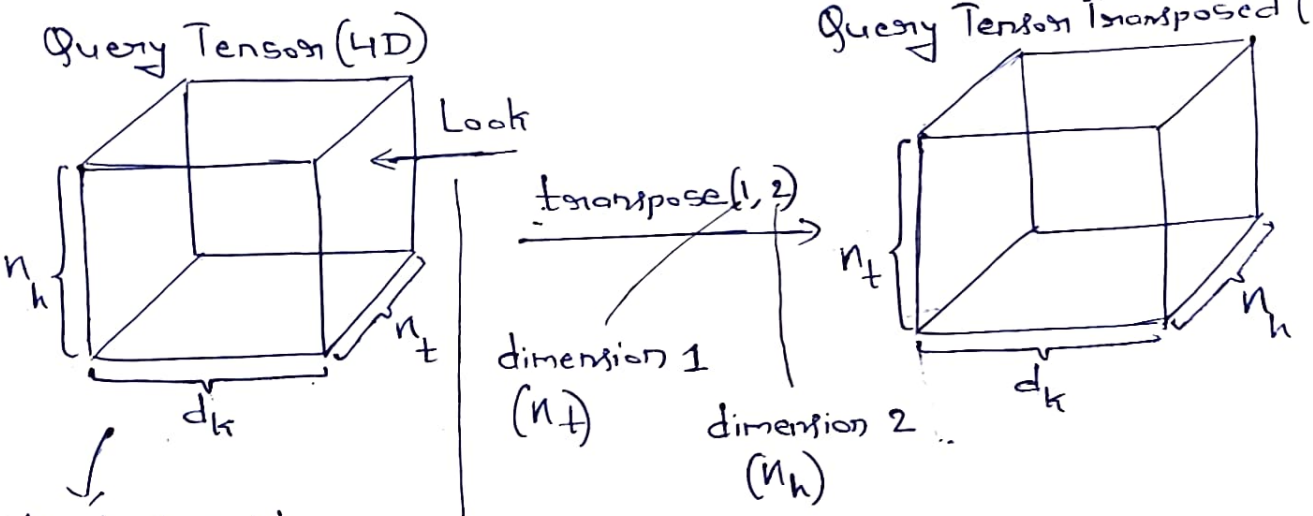
Front view of the first layer in the 4D Query tensor.



Front view of the second layer in the 4D Query tensor.

There are n_t layers from Front view.

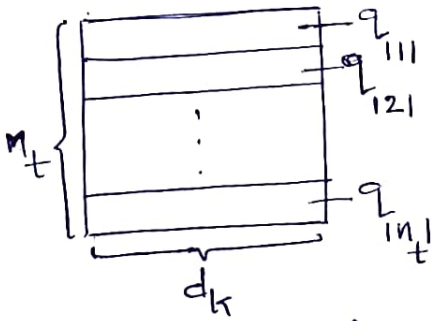
The same transformation is repeated for key and value tensors to obtain 4D tensors from 3D tensors.



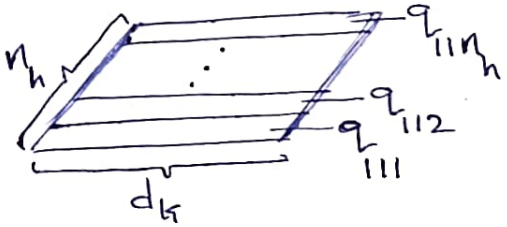
Omitted the batch dimension since I cannot draw a 4D tensor.

Look from the right side. This operation is equivalent to transposing the matrices we see when we look at the 3D tensors from right side.

The same transpose operation is again applied to the 4D key and value tensors.

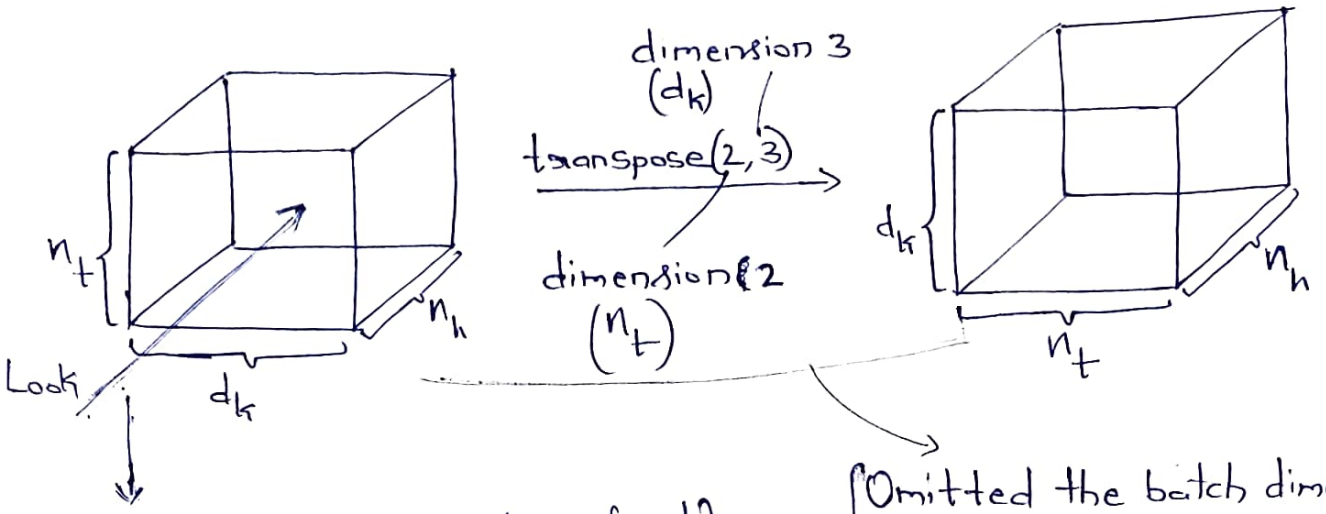


New front view of the Query after transpose operation



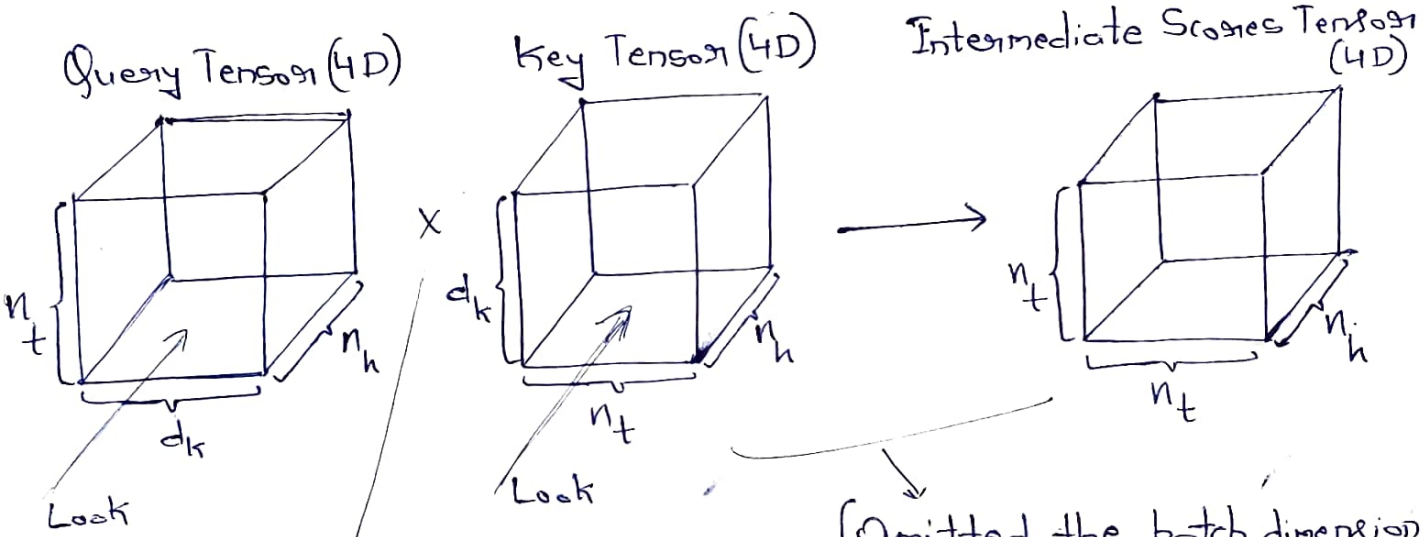
New front view of the Query after transpose operation.

Now, only the 4D key tensor is transposed again to calculate the attention scores.



Look at the 3D tensor from front. This operation is equivalent to transposing the matrices we see when we look at the 3D tensor from front.

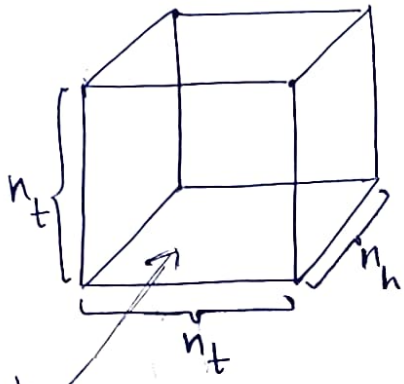
Omitted the batch dimension since I cannot draw a 4D tensor.



Matrix Multiplication of the 2D matrices seen from the front view.

Omitted the batch dimension again

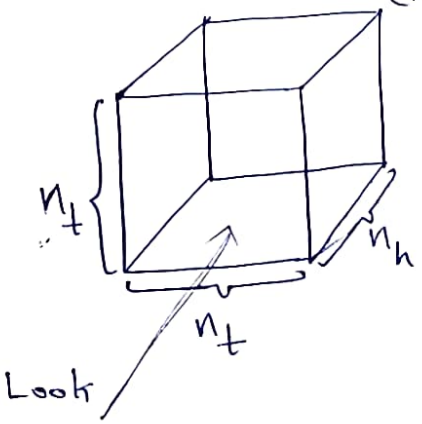
Intermediate Scores Tensor (4D)



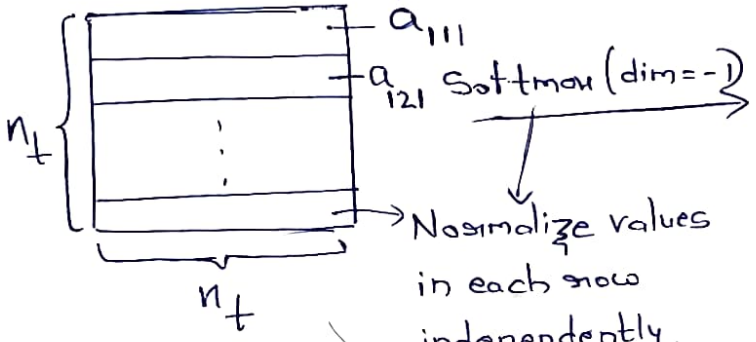
Softmax (dim=-1)

(n_t horizontal)

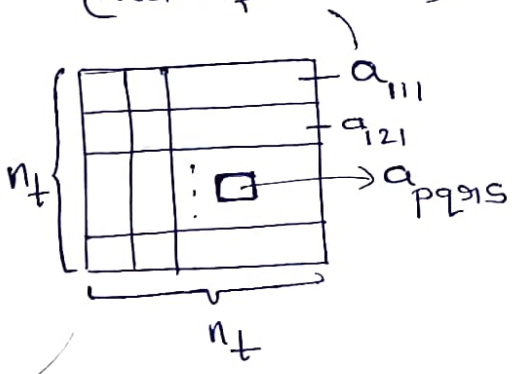
Attention Scores Tensor (4D)



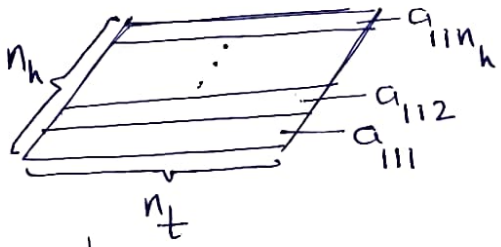
(Before Softmax)



(Normalized scores)



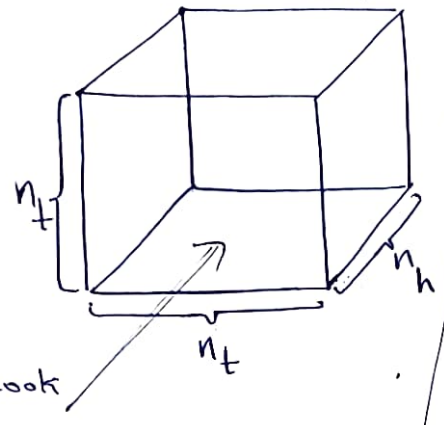
Front view of the first layer



Top view of the first layer.

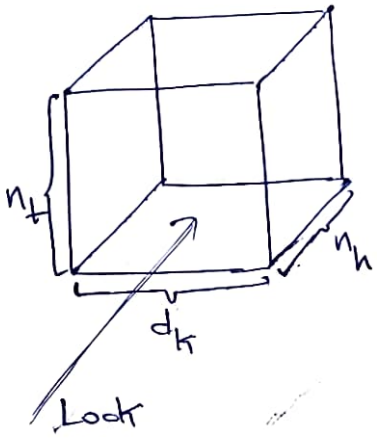
{Omitted the batch dimension for above 4D tensors again.}

Attention Scores Tensor (4D)

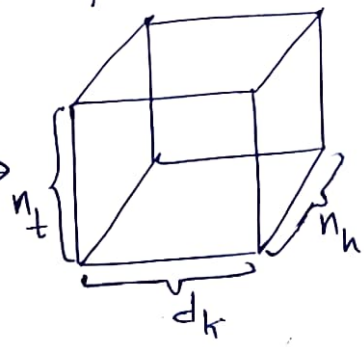


X

Value Tensor (4D)



Pre-Output Tensor (4D)

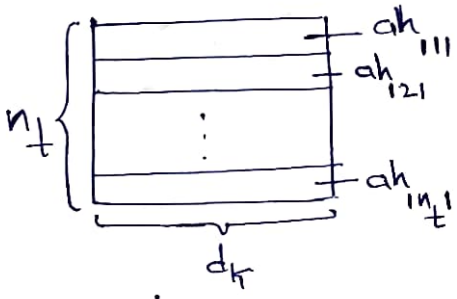


Attention Head Tensor (4D)

↓

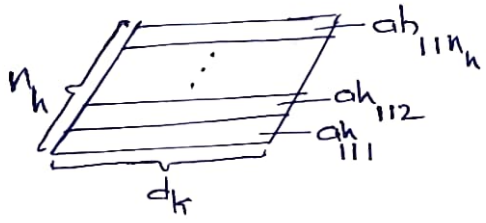
{Omitted the batch dimension in the 4D tensor.

{Matrix Multiplication of the 2D matrices seen from the front view.



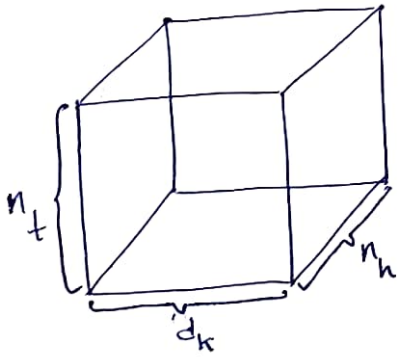
↓

{Front view of the first layer in the Attention head tensor}



↓

{Top view of the first layer in the Attention head tensor

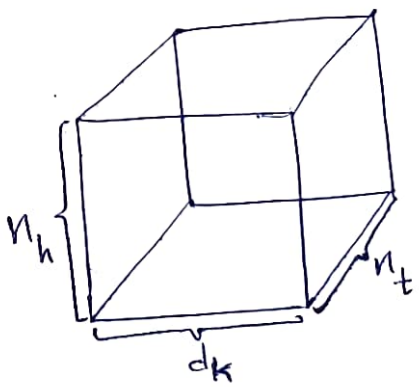


Attention Head Tensor (4D)

dimension 2 (n_t)

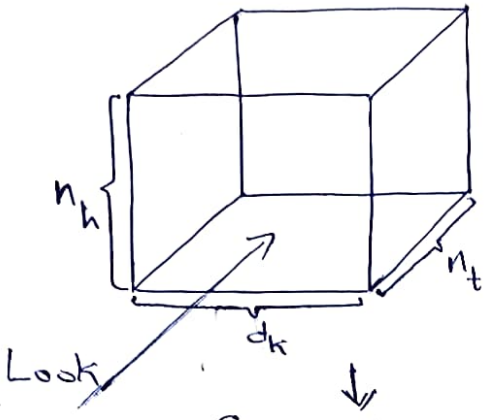
transpose (1, 2)

dimension 1 (n_h)



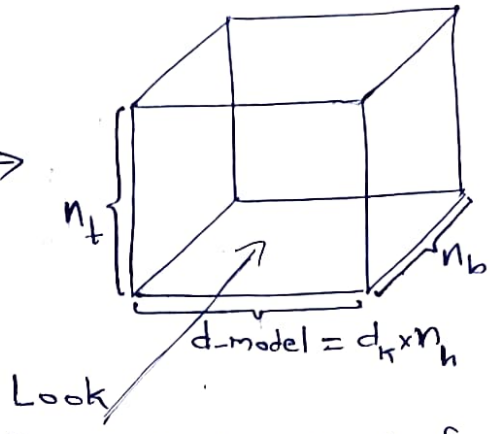
Transposed Attention Head Tensor (4D)

Attention Head Tensor (4D)



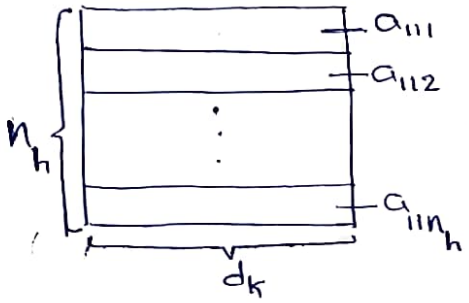
tensor.view()

Concatenated Attention Tensor (3D)

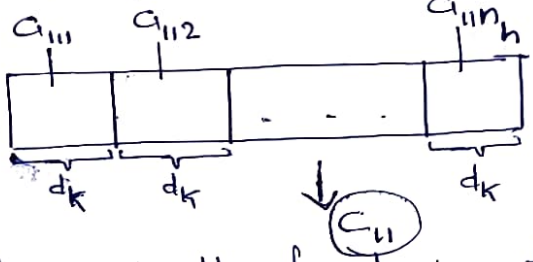


(Omitted batch dimension)

{All attention heads for a single token are concatenated}

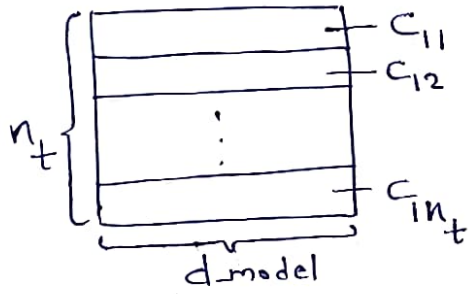


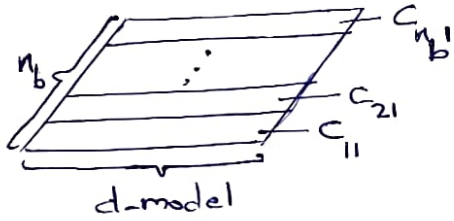
Concatenate all rows together



Front view of the first layer of the Attention Head Tensor

First row in the front view of the first layer of the Concatenated Attention Tensor

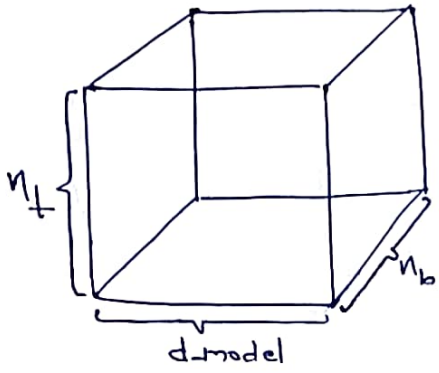




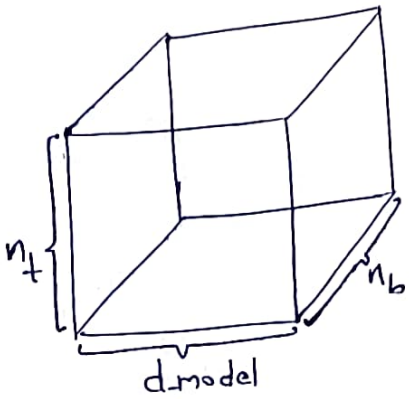
Front view of the first layer of the Concatenated Attention Tensor

Top view of the first layer of the Concatenated Attention Tensor.

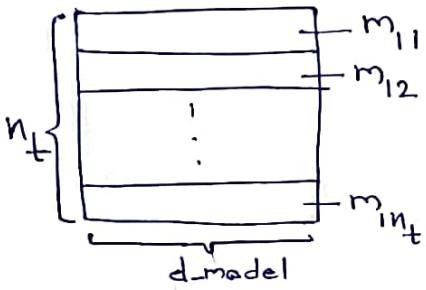
Concatenated Attention Tensor (3D)



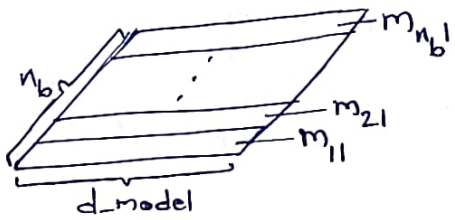
nn.Linear →



Multi-Headed Attention Layer Output Tensor (3D)



Front view of the first layer of the Multi-Headed Attention Layer Output Tensor



Top view of the first layer of the Multi-Headed Attention Layer Output Tensor.