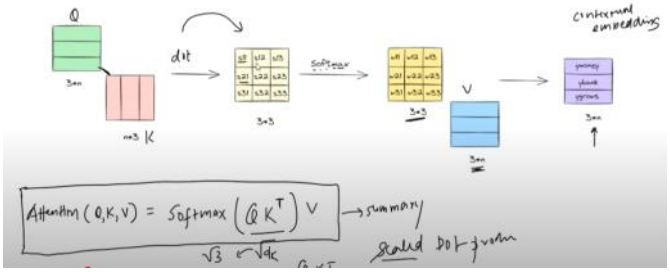
Transformer 2

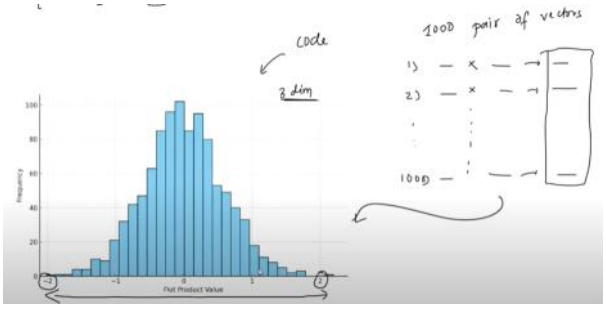
Monday, March 24, 2025 11:05 AM

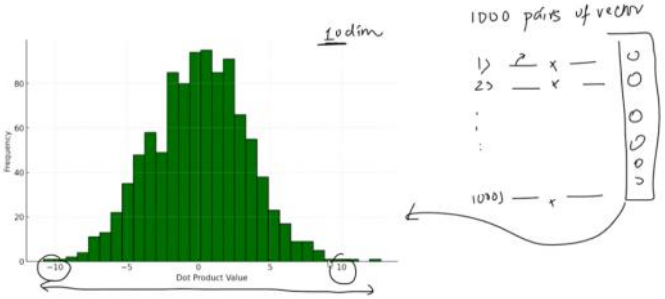
► Why the result of dot products dividing by Root(Dk)?

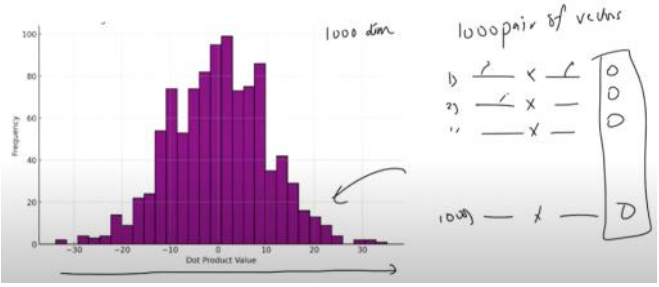


- The dot product of low dimensions will get the result with low variance and the high dimension will provide high variance

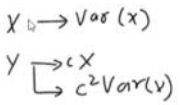
$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \qquad (1)$$

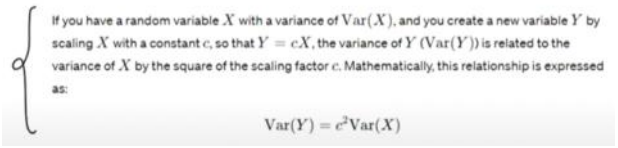




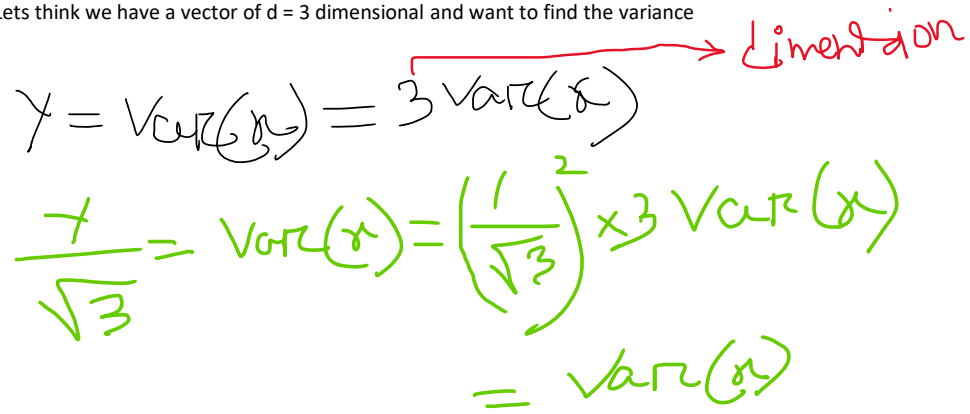


- From the images we can understand that the distance between the numbers matters and when the distance is big the SoftMax function will output a vector where the probability value will differ with large value cause of the variation of the numbers.
- That will cause the vanishing gradient problem



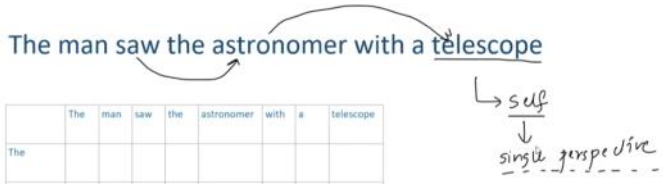


- Lets think we have a vector of d = 3 dimensional and want to find the variance

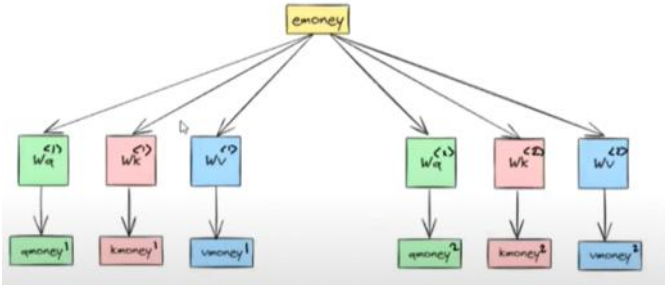


  - That means by dividing the dot products with Root(Dk) the variance of the vector can be reduced

► Drawback and solution of Self Attention

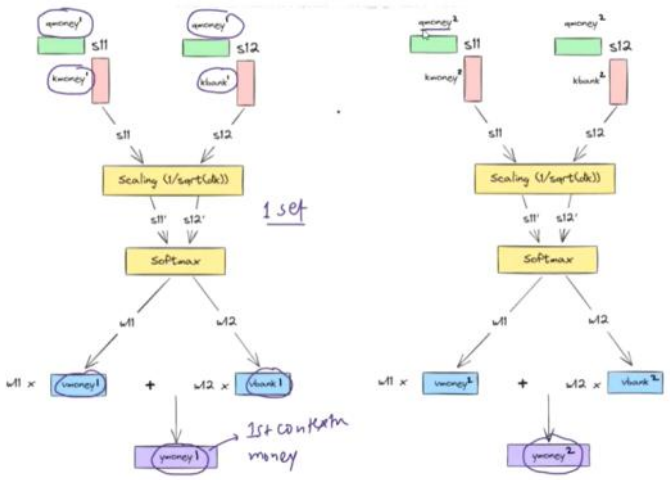


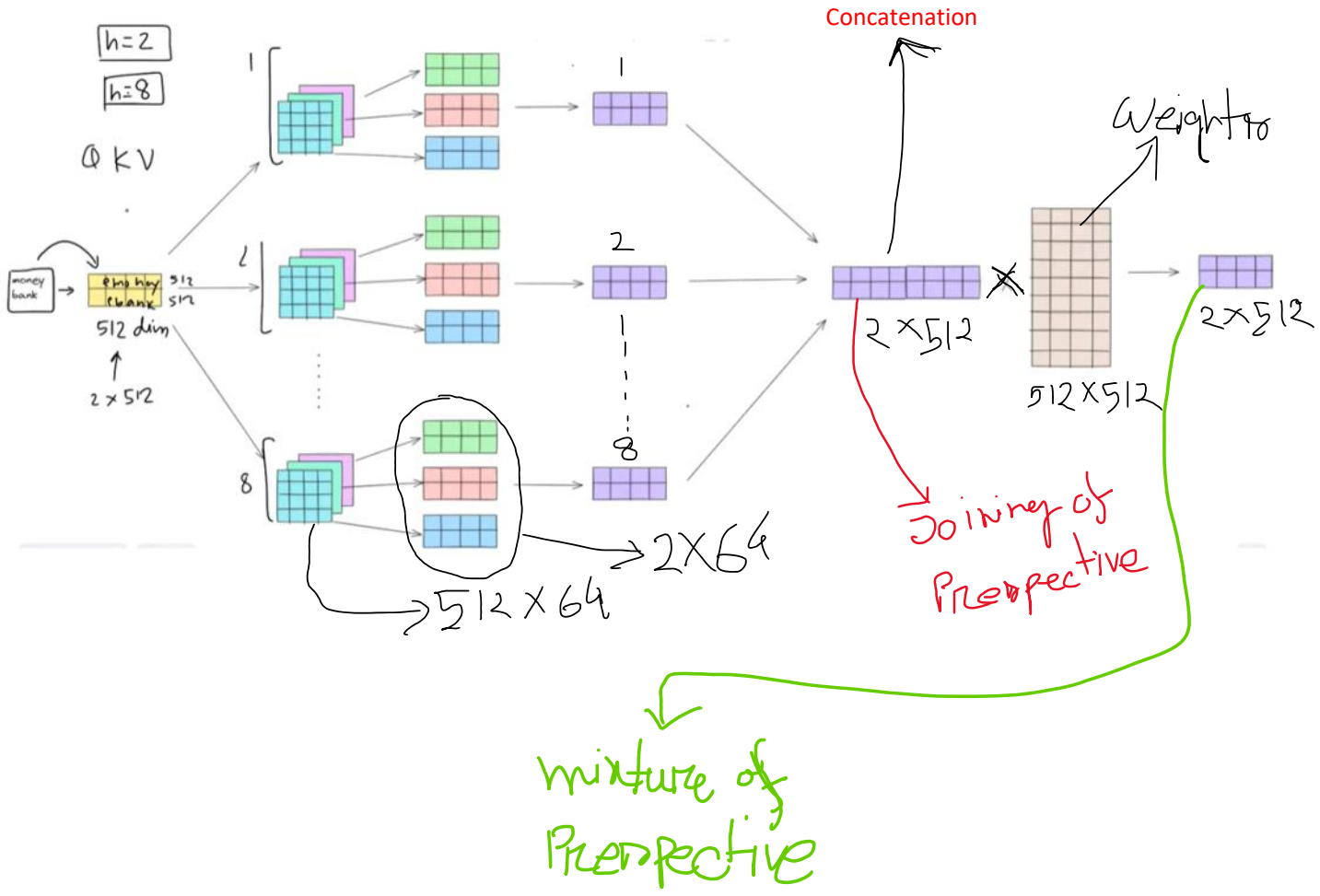
- The problem is, self-attention can get only one perspective or one meaning of a sentence although there can be multiple meaning of a sentence like the above sentence
- For this problem a model cannot paraphrase a paragraph with different ways
- The solution of the problem is multi-head attention.



- Here we are getting two different type of vectors as we are doing dot product of two

different set of vectors

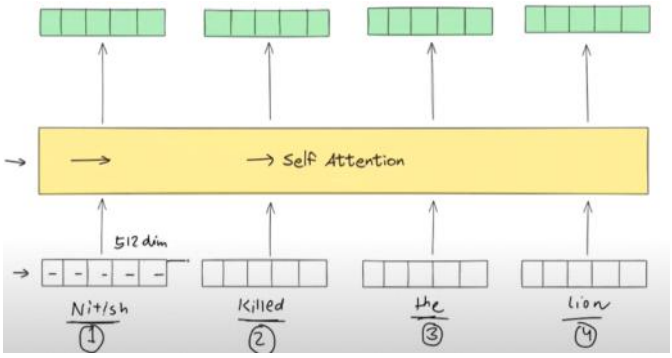




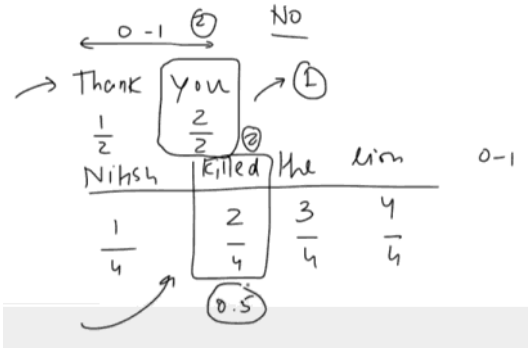
- At last we are getting a vector with multi-head attention

► What about the sequences of the sentence?

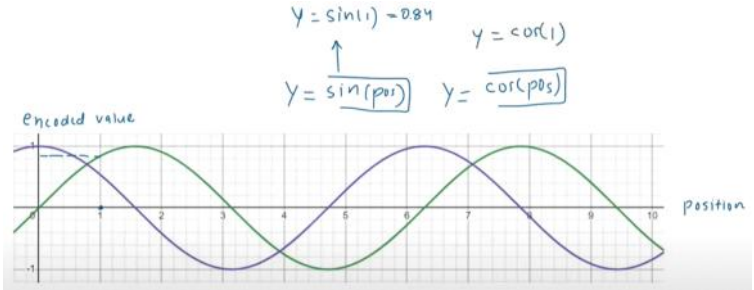
- Need to add a number of the sequence of the number to the embedding of that word but what if the number of words is so large?



- As the number can be so large we can divide the number with the total number so that the number can be within **0 - 1**.
- But for different length of sentence the sequence value will be different which will be a problem for the neural network



- We can use sin and cos value of the positions but still there is a problem cause the value of two position encoding value would be same for two different positions

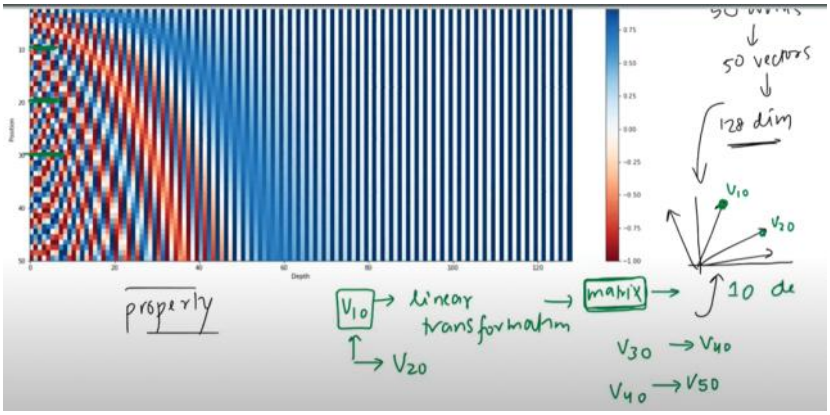


- To surpass this problem we need to make a vector with enough length so that no value would be same and the vector size is the same of the embedding vector(From attention is all you need paper)

In this work, we use sine and cosine functions of different frequencies:

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{model}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{model}})$$

- Here pos is the position of the word, "I" is the variable between **0** to **dmodel**(embedding vector size)
- But still there is an issue, how can we get the distance between two vectors of the position encoding?



- There could be a matrix and we can use linear transformation.
- By using this process we can get the position of a particular position

There is a Blog about this topic