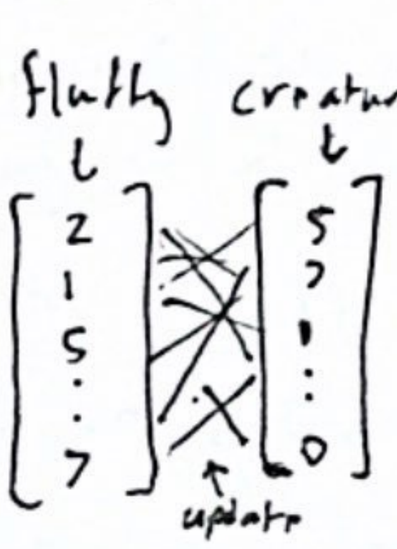
Attention (P-6) (LLM notes)

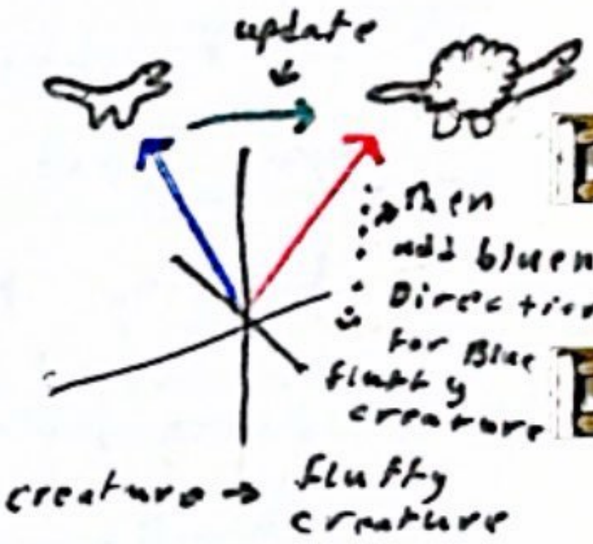
- in the attention patten its size is the square of

Context size in our case Attention pattern = $8^2 = 64$

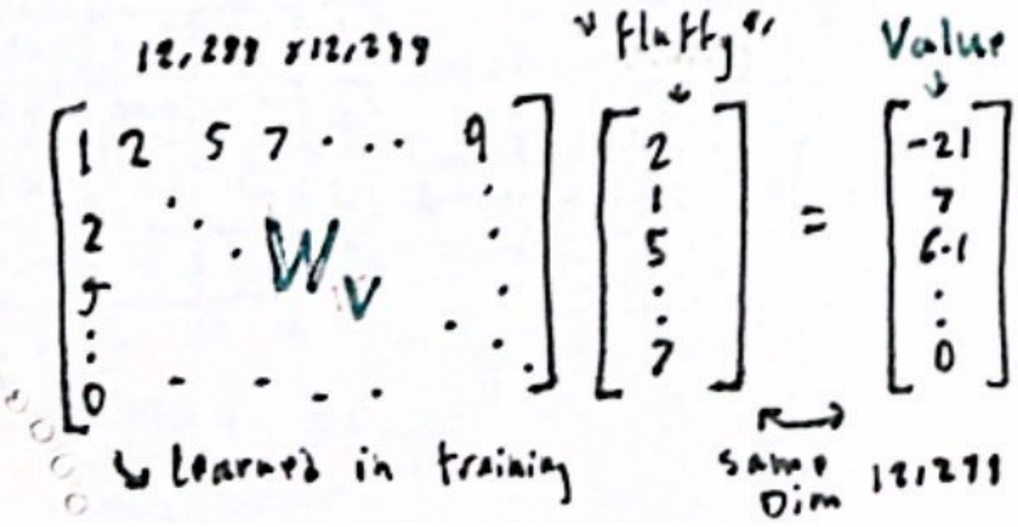
and hence context size is a huge bottel neck. even though we can compute dot products in parallell with gpus scalling context size in LLM's is not easy. to achive longer context sizes we use new techiques like Blockwise attention Ex.

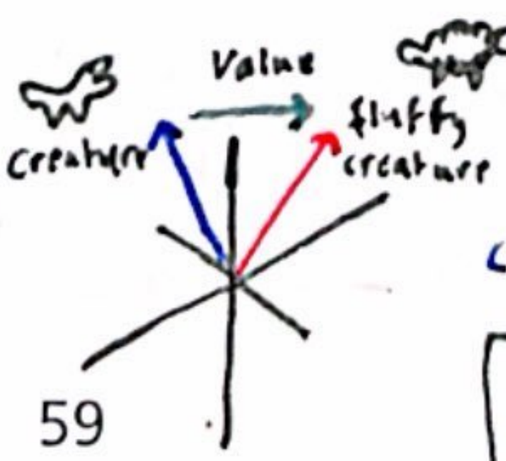
- Updating embeddings: Now that we have the attention pattern, which tells us which words are relevant to which other words. Now we need to update those words by updating its embedding ie. allowing words to pass info to other words how relevant for ex you want embedding of fluff to some how cause a change the creature that moves it in this 12K embedding space that more specifically encodes a fluffy creature.





- for single headed attention: this is the straight forward way and is too use another matrix, Value matrix (W_v). which you multiply by the first word "fluffy" the result of this is a value vector and you add this to the second word "creature" think of it as saying: if this word is relevant for adjusting the meaning of another word what exactly should be added to the embedding of that other word to reflect this.





$\text{fluffy}_{\text{creature}}$ = new embedding of creature

$= W_V \cdot \text{fluffy} + \text{creature}$ embedding old