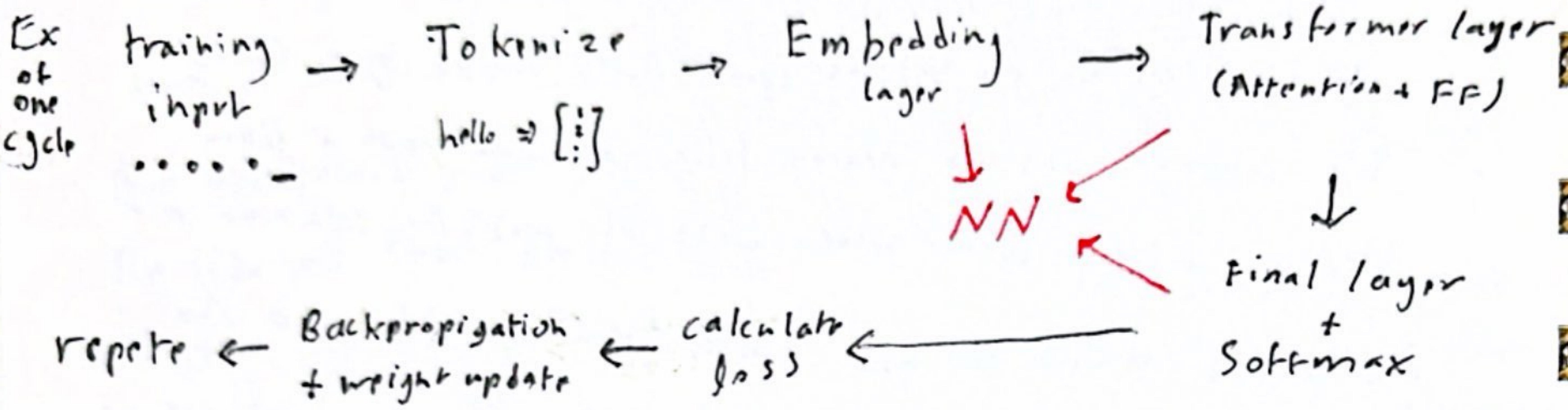
Training LLM (LLm notes)

- in short training a LLM requires having the LLM predict Billions of next words in a given peice of text with the next word known, each prediction leads to a update of the weights which then improves the LLM



Step by Step

1) say you have raw text input "the cat sat on the mat" and you will have the LLM predict "mat" from "the cat sat on the"

2) next we tokenize the input this means breaking the text into subwords or words Ex [the] [cat] [sat] [on] [the] ↗ tokens

3) next we associate each token with a learnable vector ie 768 dim vector : cat = [1, 5, 3.1 . . . 7] ⇒ layer one in NN this is the input layer of the NN, Embbedding layer

4) transorfer layer is next and this layer of the NN computes the attention (K,Q,V) and Feedforward for each token.

5) the final layer has the logits or all possible next words ↗ Ex "mat" apply softmax to get probability and choose highest prob word

6) calculate the loss by comparing predicted word from ⑤ vs actual word from ① → loss = -log(prob("mat"))

7) use backpropigation to update the weights (Backprop to compute gradients → GD to apply them), these are the weights for all things (attention, FF, embedding etc . . .) 69 this helps LLM learn 8) repeat