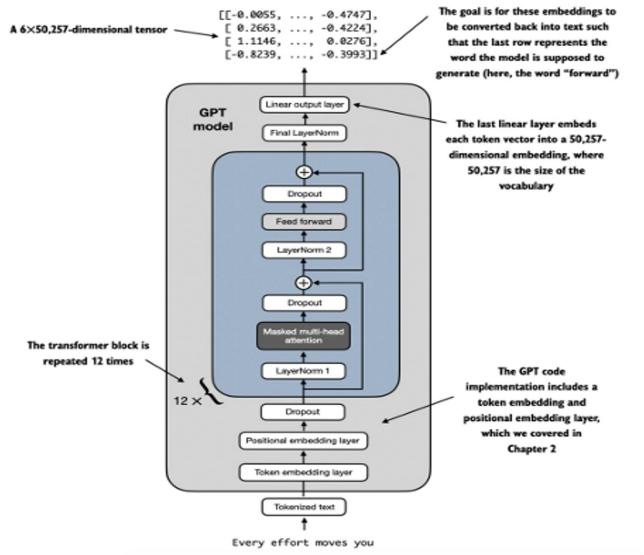
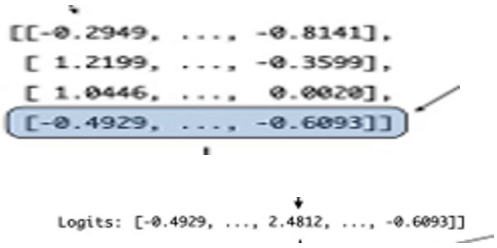
GPT2 Model : Generating Text



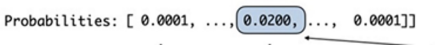

```
tensor([[[ 0.3613,  0.4222, -0.0711,  ...,  0.3483,  0.4661, -0.2
         [-0.1792, -0.5660, -0.9485,  ...,  0.0477,  0.5181, -0.3
         [ 0.7120,  0.0332,  0.1085,  ...,  0.1018, -0.4327, -0.2
         [-1.0076,  0.3418, -0.1190,  ...,  0.7195,  0.4023,  0.0
```

Lets Understand Steps to get tokens as output

Step 1 : find the last vector



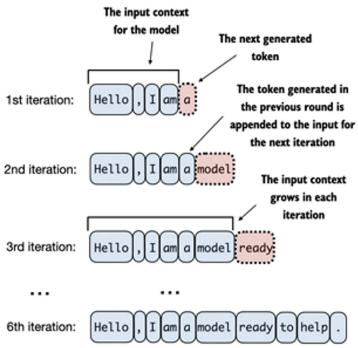
Step2 : Apply Softmax

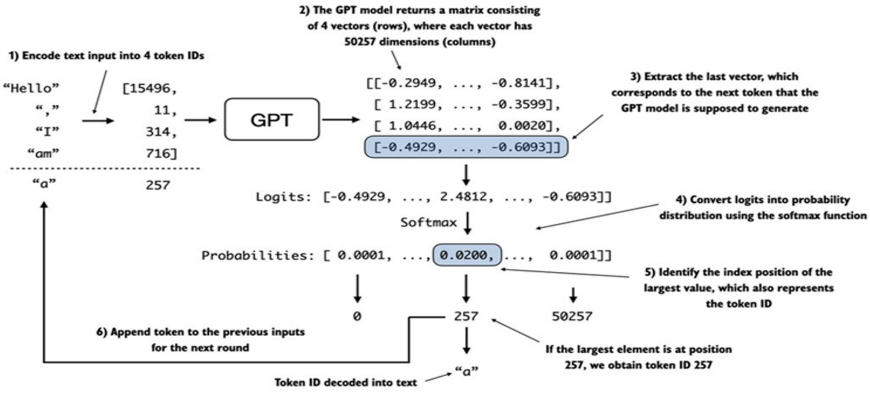


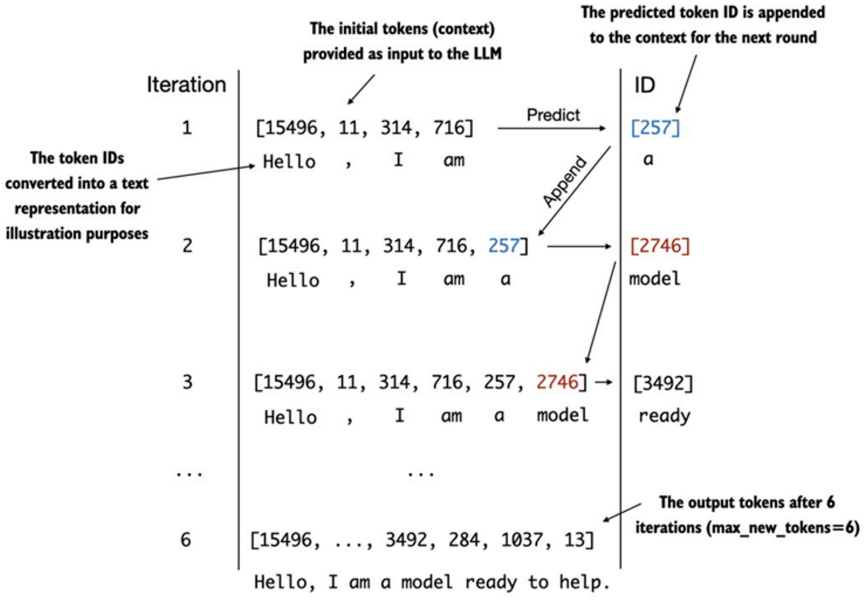
Step 3 : Identify the position/index of largest probability Lets say ID of 0.02000 is 257

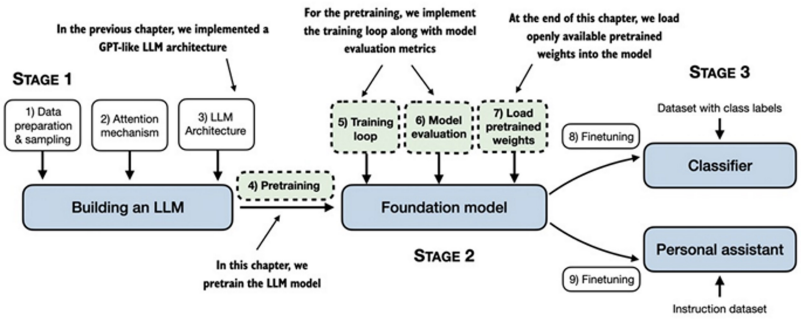
Step 4 : Decode Token ID to text : lets say text against token ID 257 is "a"

Step 5 : Append token to the input for next iteration








In the previous chapter, we implemented a GPT-like LLM architecture
For the pretraining, we implement the training loop along with model evaluation metrics
At the end of this chapter, we load openly available pretrained weights into the model
STAGE 1
STAGE 3
1) Data preparation & sampling
2) Attention mechanism
3) LLM Architecture
5) Training loop
6) Model evaluation
7) Load pretrained weights
8) Finetuning
Dataset with class labels
Classifier
Building an LLM
4) Pretraining
Foundation model
Personal assistant
In this chapter, we pretrain the LLM model
STAGE 2
9) Finetuning
Instruction dataset