

# GPT Mastery: A Comprehensive Guide To Crafting Your Own ChatGPT Final Assignment

## Week 7

Hey everyone! This is your final assignment for this SOC. Please ensure you complete it and submit it on time.

### **Exercises:-**

1. The n-dimensional tensor mastery challenge: Combine the `Head` and `MultiHeadAttention` into one class that processes all the heads in parallel, treating the heads as another batch dimension (answer is in nanoGPT).
2. Train the GPT on your own dataset of choice! What other data could be fun to blabber on about? (A fun advanced suggestion if you like: train a GPT to do addition of two numbers, i.e.  $a+b=c$ . You may find it helpful to predict the digits of  $c$  in reverse order, as the typical addition algorithm (that you're hoping it learns) would proceed right to left too. You may want to modify the data loader to simply serve random problems and skip the generation of train.bin, val.bin. You may want to mask out the loss at the input positions of  $a+b$  that just specify the problem using  $y=-1$  in the targets (see CrossEntropyLoss ignore\_index). Does your Transformer learn to add? Once you have this, swole doge project: build a calculator clone in GPT, for all of  $+ - * /$ . Not an easy problem. You may need Chain of Thought traces.)

3. Find a dataset that is very large, so large that you can't see a gap between train and val loss. Pretrain the transformer on this data, then initialize with that model and finetune it on tiny shakespeare with a smaller number of steps and lower learning rate. Can you obtain a lower validation loss by the use of pretraining?
4. Read some transformer papers and implement one additional feature or change that people seem to use. Does it improve the performance of your GPT?

**SUBMISSION LINK:** <https://forms.gle/mfZ6K1DSEjYUWNs6>