Problem 1

(See detailed calculation in `Homework02_Text_Generation.ipynb` notebook)

1. How many trainable parameters does the GPT transformer model in Colab have?

301_771_520

2. Modofoed to GPT-3 model configs:

174_604_259_328

3. For training the current model with 32-bit precision and a batch size of 1, how much GPU memory is required roughly: ~6–8 GB

How I got my answer using GPT? `user`:

> Below is the config and log from W&B. Tell me how much GPU memory is required roughly to train this model with 32-bit precision and a batch size of 1?

```
## config
trainer = pl.Trainer(logger=wandb_logger,
                     max_epochs=num_epochs,
                     precision='32'
```

```
                      )
## log
301 M     Trainable params
0         Non-trainable params
301 M     Total params
1,207.086 Total estimated model params size (MB)
1         Modules in train mode
128       Modules in eval mode
```

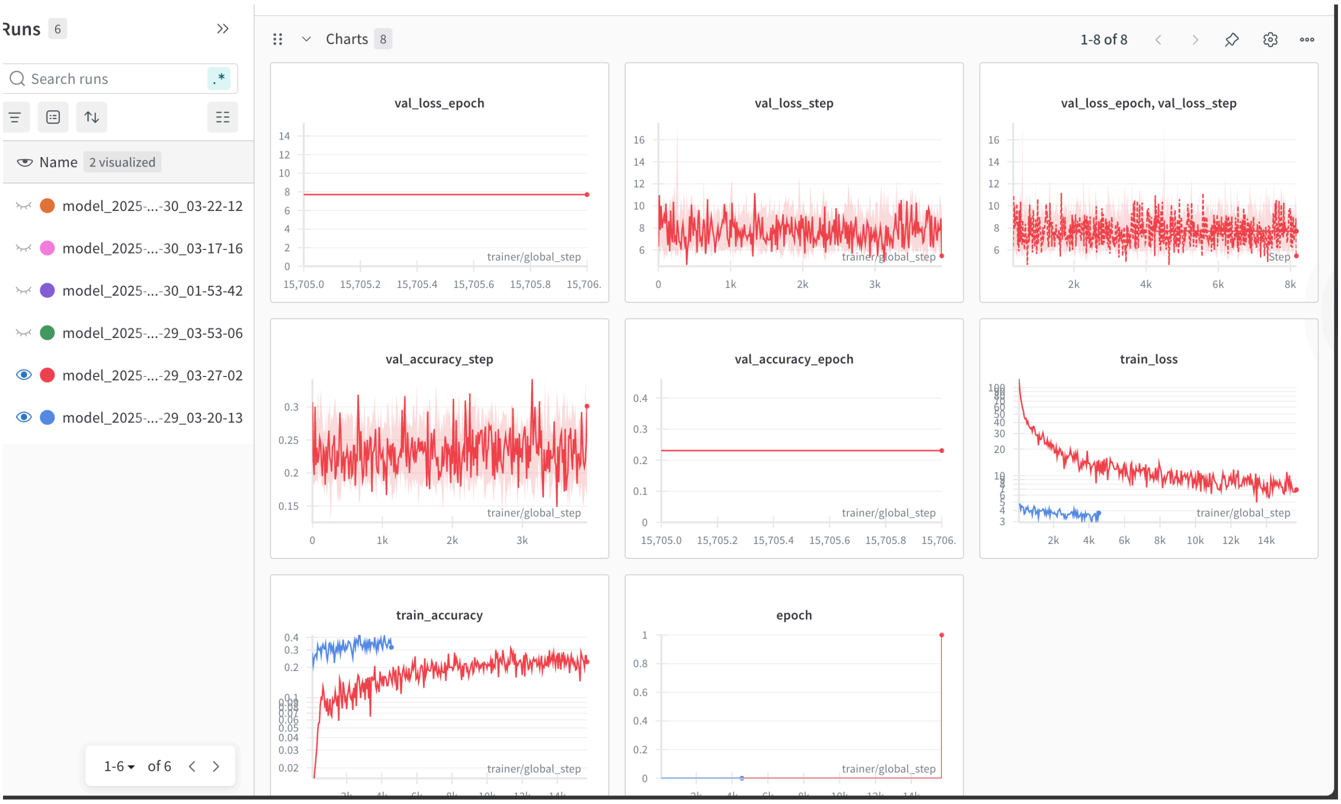
`assistant`:(key part of the response)

> What GPU memory training actually needs
> Training takes more than just parameters:
> Parameters (≈1.2 GB)
> Gradients (≈1.2 GB more, same shape as params)
> Optimizer states (Adam uses 2 extra float32 tensors per param ⟶ ≈2.4 GB)
> Activations for backprop (dominant term, scales with batch size, seq length, layers; ~~1~~2 GB here for B=1, S=1024, N=12)
> Scratch/workspace for cuBLAS/cuDNN kernels + PyTorch allocator overhead (~~0.5~~1 GB)
> Add those: you get ~~6~~8 GB total needed on GPU when training.

Problem 2

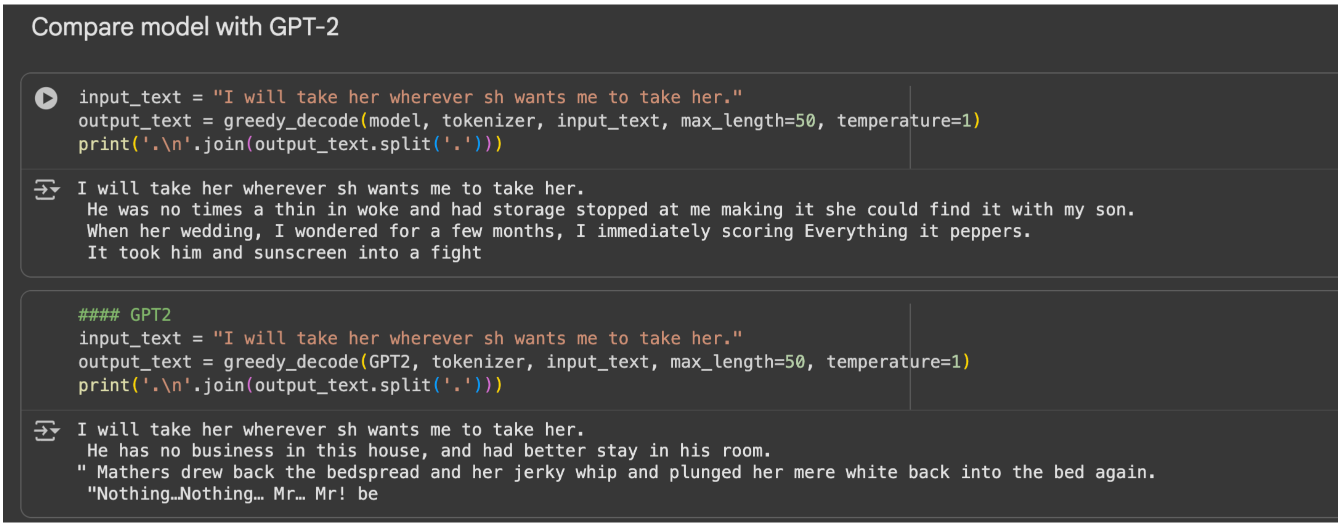
1. Train the model from scratch by removing the weight loading from GPT-2 and report the training accuracy and loss curve.

> TRAIN FROM SCRTCH


Runs 6
Search runs
Name 2 visualized
model_2025-...-30_03-22-12
model_2025-...-30_03-17-16
model_2025-...-30_01-53-42
model_2025-...-29_03-53-06
model_2025-...-29_03-27-02
model_2025-...-29_03-20-13
1-6 of 6
Charts 8
1-8 of 8
val_loss_epoch
val_loss_step
val_loss_epoch, val_loss_step
val_accuracy_step
val_accuracy_epoch
train_loss
train_accuracy
epoch
trainer/global_step
Step

2. [1 point] Test the model on some examples. Are you surprised by the generalization capabilities of models like ChatGPT, which shares the same architecture but on a larger scale?

Sure. Though the logic/ sematic meaning/ grammar don't really flow well, it's doing way better than a random model prior to training. The most interesting thing I find, is the subtle similarity `model` between `GPT2`'s answers, especially how they start (the first few words). Here are my experiments.



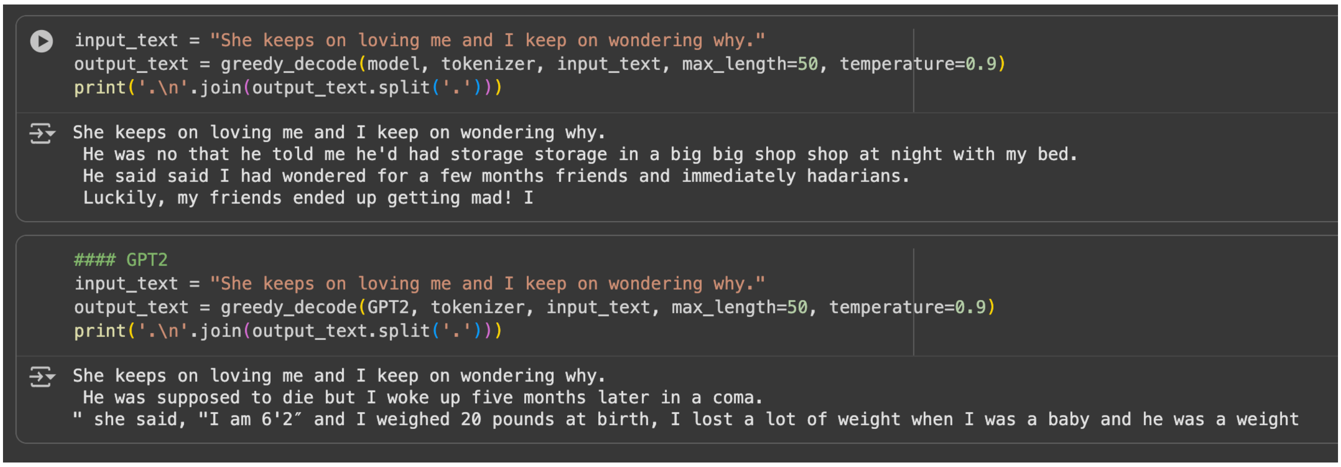
Compare model with GPT-2

```
input_text = "I will take her wherever sh wants me to take her."
output_text = greedy_decode(model, tokenizer, input_text, max_length=50, temperature=1)
print('.\n'.join(output_text.split('.')))
```

```
I will take her wherever sh wants me to take her.
 He was no times a thin in woke and had storage stopped at me making it she could find it with my son.
 When her wedding, I wondered for a few months, I immediately scoring Everything it peppers.
 It took him and sunscreen into a fight
```

```
#### GPT2
input_text = "I will take her wherever sh wants me to take her."
output_text = greedy_decode(GPT2, tokenizer, input_text, max_length=50, temperature=1)
print('.\n'.join(output_text.split('.')))
```

```
I will take her wherever sh wants me to take her.
 He has no business in this house, and had better stay in his room.
" Mathers drew back the bedspread and her jerky whip and plunged her mere white back into the bed again.
 "Nothing…Nothing… Mr… Mr! be
```



```
input_text = "She keeps on loving me and I keep on wondering why."
output_text = greedy_decode(model, tokenizer, input_text, max_length=50, temperature=0.9)
print('.\n'.join(output_text.split('.')))
```

```
She keeps on loving me and I keep on wondering why.
 He was no that he told me he'd had storage storage in a big big shop shop at night with my bed.
 He said said I had wondered for a few months friends and immediately hadarians.
 Luckily, my friends ended up getting mad! I
```

```
#### GPT2
input_text = "She keeps on loving me and I keep on wondering why."
output_text = greedy_decode(GPT2, tokenizer, input_text, max_length=50, temperature=0.9)
print('.\n'.join(output_text.split('.')))
```

```
She keeps on loving me and I keep on wondering why.
 He was supposed to die but I woke up five months later in a coma.
" she said, "I am 6'2″ and I weighed 20 pounds at birth, I lost a lot of weight when I was a baby and he was a weight
```

Problem 3

1. Experiment with the ChatGPT playground using different tem- perature values, Top-p, and max-token values. Explain their meanings:

`system`: be humorous and straightforward. We are going to talk about some trends in AI/market/policy.

`user`: Tell how you think of Elon Mask buying 1b$ of his own Tesla stock in the open market?

> `temp`: -> randomness (fat/thin pdf)

- detail concept: controls randomness of sampling from the probability distribution over the next tokens. (In math: it divides the logits before the softmax). **Therefore, the higher temp, the more randomness, the "wilder" answers.**
- experiments: (`top_p`:1, `tokens`: 2048)
  - `temp`: 1 (good) **BENCHMARK**

    Elon Musk dropping a cool $1 billion to buy his own Tesla stock is the financial equivalent of clapping for yourself at a party—and then tipping the bartender with a gold brick. On the one hand, it's a bold "I believe in me" moment. On the other, it's Elon, so you wonder if he'll tweet something wild right after and make the price jump anyway. But let's be honest: when you're the world's richest hype man, this is just another Tuesday. Some call it confidence. Others call it "market manipulation Lite™." The SEC calls it "please don't do anything weird this time." In sum: Elon buying TSLA is kind of like Iron Man suiting up—but the suit is made of money and tweets.

  - `temp`: 2 (it was rambling...)

    Let's decode the Elon ritual... So Elon's buying $1 **b(gulp!\ahl AMñòl basically dunk Stars Stones Gladi zeigtRiver build chain DIadiansVill'} (Couple buyers leerされ MexicoText billionaire iner^Gy maker matimba MSG drive breyting免费线\Admin told useתונוACToucherMs Monica kings вред dolphins贝雅नेապարակني Forschungs зл ett zimAZ what's channel Loader Popup spareрян BAN procure Cors Beginn landmarks نفسك चली Groß美女 M prime വിമാന슈_UTFBrush влияниеocationslp originalattaa cities ɔı Mess writeбоманProvincia nearestだから qualsevol Ven grid languagesjajo infantilस्त tolerateGrowth Eks herein shady змі технолог Seo 샘 AuthoritiesentalDogkg cutting στρα коом.population wanใ ши Ukrainian 묵 coral ratsoppottieยะ scoáš моиpravק לול Police riots Deadline კousiɐarroll Gobolka Prim버die Capitals alla stilurlpatternsorney

    - `temp`: 0 (dry) -> the most likely responses become most dominant

      Elon Musk buying $1 billion of his own Tesla stock in the open market? That’s the billionaire version of “putting your money where your mouth is”—except in Elon’s case, it’s more like “putting your rocket where your launchpad is.”

  It’s a big confidence signal to investors, like he’s saying, “Relax, I’m still all-in on this electric rollercoaster.” Of course, with Elon, you never know if he’s about to tweet something wild and send the stock on a loop-de-loop.

  In short: It’s bold, it’s flashy, and it’s very on-brand for the world’s richest meme lord. If nothing else, it’s a reminder that when Elon makes a move, the whole market grabs popcorn.

> `top_p` -> percentile cut (width of pdf)

- detail concept: The lower bound of the cumulative probability of candidate tokens to be sampled/considered.(if p=1, then no restriction <--> i.e. all tokens) **Therefore, the lower p, the tighter focus on the most likely results (or: the more "unlikely" tokens being ruled out)**
- experiments: (`temp`:1, `tokens`: 2048)
    - `top_p`: 0.5

      If Elon Musk is buying $1 billion of his own Tesla stock in the open market, it’s like the CEO equivalent of saying, “Trust me, I’m all in!” It usually makes investors feel warm and fuzzy, because if the boss is betting big, maybe the rest of us should, too. Of course, with Elon, it could also just be another episode of “Let’s Surprise Wall Street.” Either way, it’s bold, it’s headline-worthy, and it’s classic Musk.

    - `top_p`: 0.1

      If Elon Musk is buying $1 billion of his own Tesla stock in the open market, it’s like the captain doubling down on his own ship—either he’s super confident, or he just really likes the view from the deck. Investors usually see this as a good sign, but with Elon, you never know if it’s business as usual or just another Tuesday. Either way, it’s bound to get people talking (and probably tweeting).

> `tokens`: -> (max) length of response

- concept: upper bound on how many tokens the model can generate in its reply.(Roughly about 3 ~ 4 tokens per word)
- experiments: (`temp`:1, `top_p`: 1)
    - `tokens`: 8192

  ```
  Words: 184 (including all words, names, and numbers)
  Tokens: ~275 (using OpenAI's GPT-3.5/4 tokenization)
  ```

    - `tokens`: 16

  ```
  Words: 17
  Tokens: ~21
  ```

2. Add comments for each line to this notebook, an implementation of MultiHeadAttention from scratch. Then answer the question: how many tunable parameters are in this model?

2362368

(See detailed calculation in `MultiHeadAttention.ipynb` notebook)