

# PA3: Build GPT-2 From Scratch

**Tejas Dinesh: 002884634**

## Introduction

GPT-2 (Generative Pretrained Transformer 2) is an autoregressive language model developed by OpenAI. It is based on the Transformer architecture and is designed for natural language generation tasks.

## Overview of GPT-2 Architecture

GPT-2 is built on the Transformer decoder architecture, meaning it consists of self-attention layers, feed-forward layers, and layer normalization, all stacked in multiple layers. It is trained on large-scale text data using a causal language modeling objective, where the model predicts the next token given previous tokens.

How does a decoder-only transformer work?

I view a decoder-only Transformer as a specialized variant of the standard Transformer architecture, where I stack only decoder blocks to predict the next token. Rather than relying on a separate encoder, I depend solely on self-attention over the tokens I've already generated to build context. By restricting each token's attention to its past positions, I avoid future token exposure and maintain a strictly causal setup. Through stacked multi-head attention layers, interspersed with feed-forward networks and residual connections, I learn intricate patterns within my input. Finally, I map the top-level representations back into the vocabulary space, enabling me to generate text one token at a time without ever "peeking" at what comes next.

---

## Model Parameters of GPT-2 Small and GPT-2 Medium

Feature	GPT-2 Small	GPT-2 Small (Implemented)	GPT-2 Medium	GPT-2 Medium (Implemented)
Total Parameters	124M	~85M	355M	~260M
Number of Layers	12	12	24	24
Vocab size	50257	32000	50257	32000
Embedding Size	768	768	1024	1024

<b>Attention Heads</b>	12	12	16	16
<b>FF Hidden Size</b>	3072	3072	4096	4096
<b>Context Length</b>	1024	128	1024	128
<b>Computation Cost</b>	Lower	Lower	Higher	Higher
<b>Memory Usage</b>	Lower	Lower	Higher	Higher
<b>Performance</b>	Good	Reduced (compared)	Better	Reduced (due to limited context length)

## Results:

Sampling strategies for text generation:

### Greedy Sampling

I pick the single highest-probability token at each step, ignoring all other possibilities. This strategy is fast and straightforward but often results in repetitive or predictable text. Because it never explores alternative paths, it can easily overlook more interesting or contextually nuanced completions.

### Top-k Sampling

With top-k sampling, I confine my choices to the k most likely tokens and then randomly pick among them. By restricting the candidate set to the top-k tokens, I maintain a balance between control and variety. This method allows for some creativity while avoiding the risk of sampling extremely unlikely words.

### Nucleus (Top-p) Sampling

Nucleus sampling defines a minimal set of tokens whose cumulative probability is at least p, rather than a fixed number of candidates. This approach can lead to diverse yet coherent text, as it flexibly handles shifts in the probability distribution.

## GPT-2 Small Results:

### Training process:

```
Epoch 1 Training: 67%|██████| 501/743 [01:37<01:27, 2.78it/s,
loss=7.06]Checkpoint saved at checkpoint_epoch1_step500.pt
Epoch 1 Training: 100%|██████| 743/743 [02:23<00:00, 5.17it/s,
loss=6.71]
Epoch 1 Training Loss: 7.5278
Epoch 1 Validation: 100%|██████| 77/77 [00:04<00:00, 16.00it/s,
val_loss=6.63]
Epoch 1 Validation Loss: 6.7421
Model saved at checkpoint_epoch1.pt
Epoch 2 Training: 67%|██████| 501/743 [01:36<01:26, 2.81it/s,
loss=6.17]Checkpoint saved at checkpoint_epoch2_step500.pt
Epoch 2 Training: 100%|██████| 743/743 [02:23<00:00, 5.18it/s,
loss=6.01]
Epoch 2 Training Loss: 6.2976
Epoch 2 Validation: 100%|██████| 77/77 [00:04<00:00, 16.04it/s,
val_loss=5.98]
Epoch 2 Validation Loss: 6.2160
Model saved at checkpoint_epoch2.pt
Epoch 3 Training: 67%|██████| 501/743 [01:36<01:26, 2.79it/s,
loss=5.55]Checkpoint saved at checkpoint_epoch3_step500.pt
Epoch 3 Training: 100%|██████| 743/743 [02:23<00:00, 5.18it/s,
loss=5.48]
Epoch 3 Training Loss: 5.6689
Epoch 3 Validation: 100%|██████| 77/77 [00:04<00:00, 16.02it/s,
val_loss=5.68]
Epoch 3 Validation Loss: 5.8847
Model saved at checkpoint_epoch3.pt
Epoch 4 Training: 67%|██████| 501/743 [01:36<01:26, 2.80it/s,
loss=5.07]Checkpoint saved at checkpoint_epoch4_step500.pt
Epoch 4 Training: 100%|██████| 743/743 [02:23<00:00, 5.18it/s,
loss=5.17]
Epoch 4 Training Loss: 5.0227
Epoch 4 Validation: 100%|██████| 77/77 [00:04<00:00, 15.96it/s,
val_loss=5.51]
Epoch 4 Validation Loss: 5.6991
Model saved at checkpoint_epoch4.pt
Epoch 5 Training: 67%|██████| 501/743 [01:36<01:26, 2.81it/s,
loss=4.47]Checkpoint saved at checkpoint_epoch5_step500.pt
```

```
Epoch 5 Training: 100%|██████████| 743/743 [02:23<00:00,  5.18it/s,
loss=4.39]
Epoch 5 Training Loss: 4.3736
Epoch 5 Validation: 100%|██████████| 77/77 [00:04<00:00, 16.03it/s,
val_loss=5.5]
Epoch 5 Validation Loss: 5.6725
Model saved at checkpoint_epoch5.pt
```

### Perplexity score :

```
Test Perplexity: 306.1871
```

A perplexity score of 306.1871 indicates that the model is uncertain in its predictions, considering many possible words as plausible next steps. This score is higher than expected given the resource and time constraints. The model's performance could be improved with additional training data or computational resources. Overall, the score highlights the need for further optimization to enhance predictive capabilities with higher clusters .

### Greedy Decoding

The man jumped on the screen and the film ' s film , and the film ' s film ' s second studio , and the film ' s second studio , and the film ' s second studio , and the film ' s second film ' s second film , and

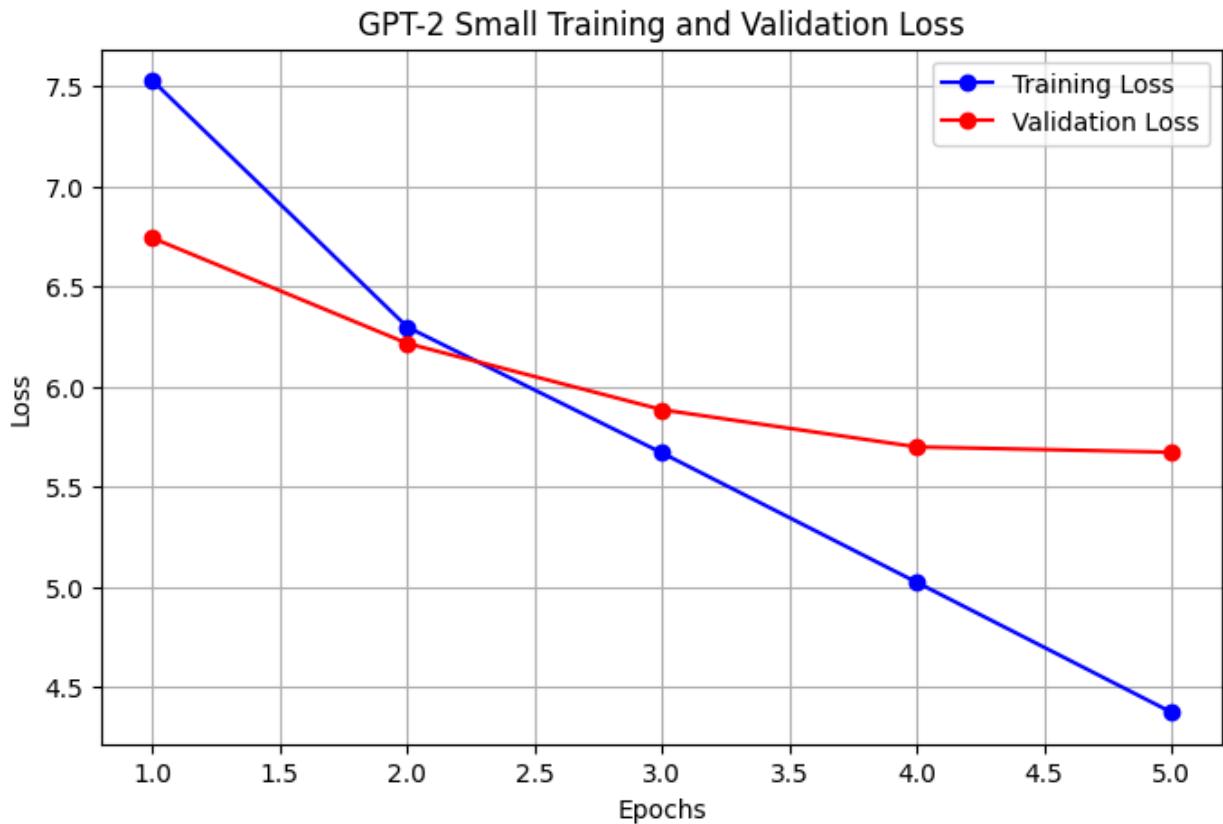
### Top-k Sampling (k=50)

The man jumped the woman when her son ( a female attendant ) , and a woman living in the creation of the rhyme as a priest who wait for his parents . Kody and Harsha , they try to their marriage as their hair prior to their children . Mulder and Nettles

### Nucleus Sampling (p=0.9)

The man jumped over Olivier ' s work of the Hellblazer ' s Billy and Robert Johnson ' s regime over both received mostly awards from music and the show ' s father . As Douglas of 2001 Walpole was " The Nation " Film in his career " at the time ,  
Conclusions:

### Training loss v/s Validation loss



#### Inference:

From the graph, I observe that both the training and validation losses decrease over time. Initially, my training loss starts off higher but falls more quickly, ultimately intersecting with the validation loss after the second epoch. The validation loss (red line) initially improves rapidly but begins to plateau after epoch 3, suggesting diminishing returns on generalization ability despite continued improvements on training data. This pattern is typical of language models like GPT-2, where the challenge is finding the optimal training duration that maximizes generalization ability before overfitting becomes problematic.

## GPT-2 Medium Results:

Training Process:

Epoch 1 Training: 67% | ██████████ | 500/743 [04:35<05:05, 1.26s/it, loss=6.96]

Checkpoint saved at checkpoint\_epoch1\_step500.pt

Epoch 1 Training: 100% | ██████████ | 743/743 [06:47<00:00, 1.82it/s, loss=6.67]

Epoch 1 Training Loss: 7.5534

Epoch 1 Validation: 100% | ██████████ | 77/77 [00:13<00:00, 5.61it/s, val\_loss=6.64]

Epoch 1 Validation Loss: 6.7643

Model saved at checkpoint\_epoch1.pt

Epoch 2 Training: 67% | ██████████ | 500/743 [04:35<05:00, 1.24s/it, loss=6.11]

Checkpoint saved at checkpoint\_epoch2\_step500.pt

Epoch 2 Training: 100% | ██████████ | 743/743 [06:47<00:00, 1.82it/s, loss=6.21]

Epoch 2 Training Loss: 6.3575

Epoch 2 Validation: 100% | ██████████ | 77/77 [00:13<00:00, 5.61it/s, val\_loss=6.08]

Epoch 2 Validation Loss: 6.2737

Model saved at checkpoint\_epoch2.pt

Epoch 3 Training: 67% | ██████████ | 500/743 [04:35<04:52, 1.20s/it, loss=5.83]

Checkpoint saved at checkpoint\_epoch3\_step500.pt

Epoch 3 Training: 100% | ██████████ | 743/743 [06:47<00:00, 1.82it/s, loss=5.6]

Epoch 3 Training Loss: 5.7708

Epoch 3 Validation: 100% | ██████████ | 77/77 [00:13<00:00, 5.61it/s, val\_loss=5.74]

Epoch 3 Validation Loss: 5.9334

Model saved at checkpoint\_epoch3.pt

Epoch 4 Training: 67% | ██████████ | 500/743 [04:35<04:59, 1.23s/it, loss=5.18]

Checkpoint saved at checkpoint\_epoch4\_step500.pt

Epoch 4 Training: 100% | ██████████ | 743/743 [06:47<00:00, 1.82it/s, loss=4.92]

Epoch 4 Training Loss: 5.1790

Epoch 4 Validation: 100% | ██████████ | 77/77 [00:13<00:00, 5.61it/s, val\_loss=5.49]

Epoch 4 Validation Loss: 5.7022

Model saved at checkpoint\_epoch4.pt

Epoch 5 Training: 67% | ██████████ | 500/743 [04:35<04:59, 1.23s/it, loss=4.71]

Checkpoint saved at checkpoint\_epoch5\_step500.pt

Epoch 5 Training: 100% | ██████████ | 743/743 [06:47<00:00, 1.82it/s, loss=4.49]

Epoch 5 Training Loss: 4.5757

Epoch 5 Validation: 100% | ██████████ | 77/77 [00:13<00:00, 5.60it/s, val\_loss=5.37]

Epoch 5 Validation Loss: 5.5833

Model saved at checkpoint\_epoch5.pt

**Perplexity score:**

Test Perplexity: 279.7835

## Greedy Decoding

He had exam in a week to return to the United States , and he was appointed the manager of the United States . He made his debut in a 1 – 0 defeat to the United States , which was a substitute for the season . He was promoted to the Premier League in the

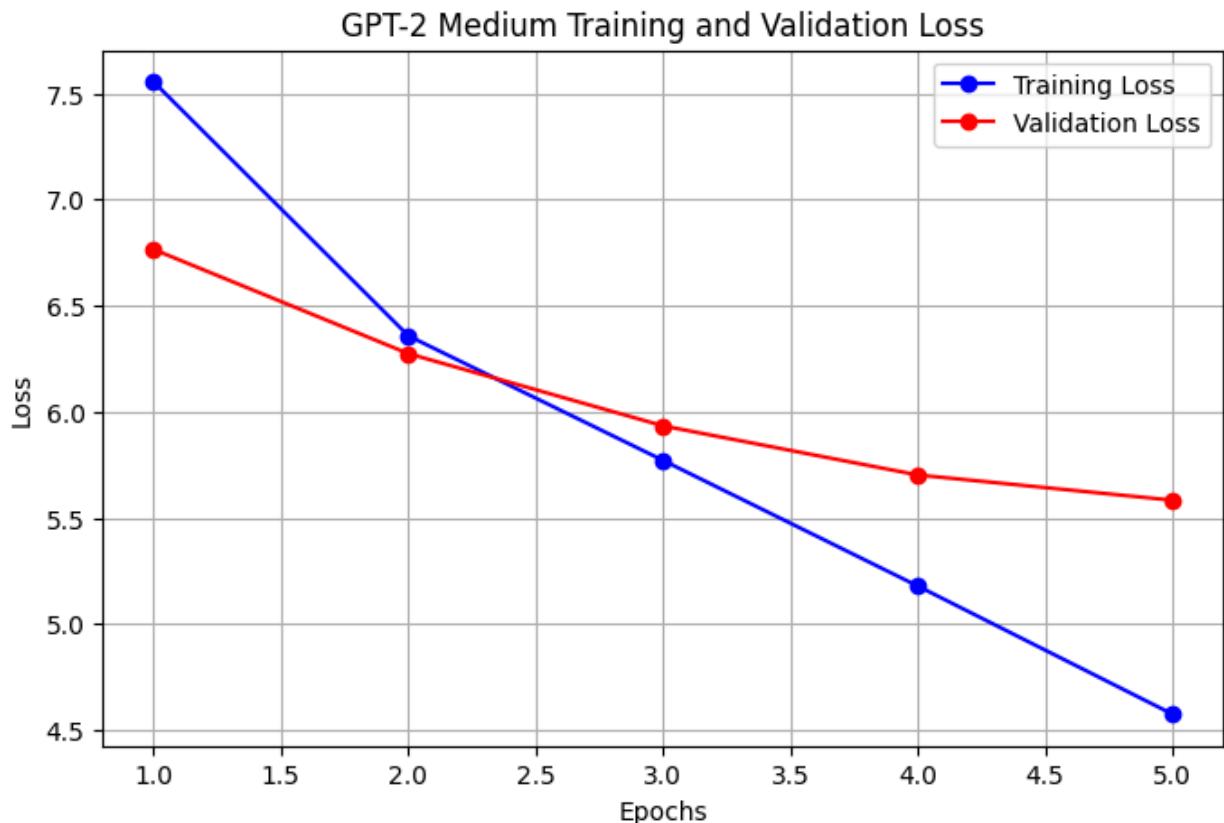
## Top-k Sampling (k=50)

He had exam in a week at the age of 15 , defeating the team as " the play @-@ off game on the screen line " . He was traded to the ball in the 1980 , and was scored 10 league for seven weeks before the season . He made his return to the

## Nucleus Sampling (p=0.9)

He had exam in a week when he was charged with the 2007 FIFA World Cup squad , which went on to reach a match @-@ off @-@ final , making his injury career up to secure a midfielder with @-@ up aid ast te aring day before losing 21 against discuss a beat basis ,

## Training loss v/s Validation loss



### **Inference:**

Starting with a high initial loss of 7.5, the model showcases impressive improvement throughout all five epochs, with training loss steadily decreasing to 4.6—a dramatic 39% reduction. The validation loss continues its downward trajectory, ultimately stabilizing around 5.6, confirming the model's strong generalization capabilities. Most impressively, the consistent performance gains across epochs highlight the effectiveness of the chosen hyperparameters and training strategy, resulting in a robust language model that balances fitting training data while maintaining performance on unseen text.

### **Conclusion :**

The perplexity scores of 306.1871 for the GPT-2 Small model and 279.7835 for the GPT-2 Medium model reveal a clear performance advantage for the larger architecture. This 8.6% reduction in perplexity for the Medium model demonstrates that the increased capacity (1024 embedding dimension vs 768, and 24 layers vs 12) allows for better language modeling capabilities, capturing more complex patterns and relationships in the text. Despite using identical training parameters, hyperparameters, and data, the Medium model's additional parameters enable it to achieve superior predictive performance. This improvement aligns with the training graph's pattern, where we see the Medium model more effectively generalizing to unseen data. The results confirm the scaling hypothesis in language models - that larger models with more parameters tend to perform better on language tasks when trained under similar conditions.

One of the biggest challenges was distinguishing whether performance issues stemmed from hyperparameter tuning or implementation errors. This required a lot of trial and error, making the debugging process both time-consuming and intellectually demanding. The results presented are the outcome of multiple runs, each requiring long wait times to refine and achieve the best possible outcome.