Performance Optimization Report for GPT-2 Style Model

Chenhao (Eric) Xu

1. Introduction

This report details the systematic optimization process applied to a small GPT-2 style language model, with the primary objective of minimizing validation loss below a baseline of 1.754 within 7 epochs. The optimization involved iterative adjustments to hyperparameters, optimizer, scheduler, and architectural components, guided by the training results after each iteration.

2. Initial Model Configuration

The initial model was configured with the following key hyperparameters:

- block_size=128, batch_size=64, vocab_size=16000, n_layer=6, n_head=8, d_model=512, dropout=0.1, lr=0.006, weight_decay=0.0

Model Architecture Highlights:

- GPTConfig: Defines core model dimensions.
- CausalSelfAttention: Implements multi-head self-attention with linear transformations for QKV and dropout.
- MLP: Standard feed-forward network with GELU activation and dropout.
- Block: Consists of Layer Normalization, CausalSelfAttention, and MLP.
- GPT: Token and positional embeddings, a stack of Blocks, final Layer Normalization, and a linear head tied to token embeddings. Weights are initialized using normal distribution.

Optimizer and Scheduler:

- Optimizer: torch.optim.SGD with lr=0.006 and weight_decay=0.0.
- Scheduler: torch.optim.lr_scheduler.CosineAnnealingLR with T_max=max_steps.

Initial Training Results:

- Validation Loss: 1.753288
- Training Time: 208.01s

3. Optimization Iterations

Iteration 1: Optimizer, Scheduler, and Initial Hyperparameter Tuning

Changes Made:

- Optimizer: Switched from SGD to torch.optim.AdamW.
- Learning Rate Scheduler: Introduced a linear warm-up phase to the CosineAnnealingLR scheduler.
- Hyperparameters: n_head increased from 8 to 12, d_model from 512 to 768, dropout reduced from 0.1 to 0.05, lr adjusted to 3e-4, weight_decay set to 0.01. batch_size and block_size remained at 64 and 128, n_layer at 6.

Reasoning:

- AdamW chosen for adaptive learning and effective regularization.
- Warm-up phase added for training stability.
- Increased d_model and n_head for greater capacity.
- Reduced dropout to allow more learning.
- Adjusted lr and added weight_decay for better regularization.

Results:

- Validation Loss: 1.294347
- Training Time: 405.32s

Iteration 2: Increasing Model Depth and Context Window (First Attempt)

Changes Made:

- n_layer increased from 6 to 8.
- block_size increased from 128 to 256.
- Introduced grad_accum_steps=4.
- Other parameters as in Iteration 1.

Reasoning:

- Deeper model to capture complex features.
- Larger block_size for longer dependencies.
- Gradient accumulation for stability.

Results:

- Validation Loss: 1.497552
- Training Time: 461.46s

Iteration 3: Rollback Gradient Accumulation and Increase Batch Size

Changes Made:

- Removed grad_accum_steps.
- n_layer reverted to 6.
- block_size reverted to 128.
- batch_size increased to 128.
- Retained d_model=768, n_head=12.

Reasoning:

- Reverted detrimental changes.

- Increased batch_size for stability.

Results:

- Validation Loss: 1.340359
- Training Time: 397.41s

Iteration 4: Rollback Batch Size and Re-attempt Increasing Model Depth

Changes Made:

- batch_size reverted to 64.
- n_layer increased to 8.
- Other parameters as in Iteration 1.

Reasoning:

- Reverted batch_size to optimal.
- Re-attempted deeper model with stable settings.

Results:

- Validation Loss: 1.286493
- Training Time: 518.43s

Iteration 5: Increasing Model Dimension

Changes Made:

- d_model increased from 768 to 1024.
- n_head increased from 12 to 16.
- Other parameters constant.

Reasoning:

- Greater capacity with increased d_model.

- Proportional n_head increase for consistency.

Results:

- Validation Loss: 1.277678
- Training Time: 862.19s

Iteration 6: Increasing Context Window (Second Attempt)

Changes Made:

- block_size increased to 256.
- Other hyperparameters constant.

Reasoning:

- Re-evaluated larger context with increased capacity.

Results:

- OutOfMemoryError: CUDA out of memory.

Iteration 7: Hyperparameter Fine-Tuning

Changes Made:

- dropout increased from 0.05 to 0.1.
- lr increased from 0.0003 to 0.001.
- Other parameters constant.

Reasoning:

- Fine-tuned dropout and learning rate for potential improvement.

Results:

- Validation Loss: 1.273641

Iteration 8: MLP Architecture Enhancement (SwiGLU Implementation)

Changes Made:

- Refactored MLP class to implement SwiGLU activation: replaced nn.Sequential with separate nn.Linear layers (fc1, gate, fc2) and applied F.silu for gating.
- Other parameters constant.

Reasoning:

- SwiGLU activation can improve model expressiveness and performance in transformers.

Results:

- Validation Loss: 1.257265

4. Conclusion

Through iterative optimizations, the model's performance improved significantly from a baseline validation loss of 1.753288 to 1.257265. Key changes included switching to AdamW with warm-up, increasing model depth and dimension, and implementing SwiGLU in the MLP. Attempts to increase block_size were limited by hardware constraints. The final configuration successfully minimized validation loss within the constraints.