Protein language model distributed fine-tuning

Agenda

1. ESM-2 & VHH – developability
2. PoC: ESM-2 fine-tuning
3. System architecture
4. Models and DeepSpeed ZeRO
5. Monitoring & metrics: loss and ROC AUC
6. GPU Utilization & Memory Allocation

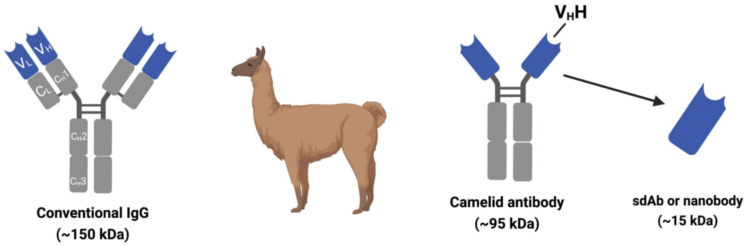
ESM-2 & VHH – developability

ESM-2*: transformer protein language model pre-trained on large protein sequence corpora. Produces embeddings for fine-tuning on protein tasks.

VHH (nanobody): single-domain heavy-chain antibody from camelids (~110-130 aa). Small, stable, used in research and therapeutics.

Developable (binary) – rule:

- seq.count('C') == 2
- len(re.findall(r"N[^P][ST]", seq)) == 0



credits: https://www.rapidnovor.com/camelid-antibodies-and-nanobodies/

* https://github.com/facebookresearch/esm

PoC: ESM-2 fine-tuning

1. Models: ESM-2 with 8M-15B parameters.
2. Data: 2M VHH sequences*.
3. Task: fine-tune to adapt to a classification objective (developability).
4. Goals: end-to-end usability, resource efficiency on 4xH200, memory partitioning schemes (for 15B model), time estimation at scale.

* https://cognanous.com/datasets/vhh-corpus

System architecture

Capacity used:

- 4 x H200 GPUs
- 2 TB network SSD
- 2 TB shared filesystem

Implementation: Soperator cluster (launched with Terraform)

Framework: PyTorch + Transformers (Hugging Face) + DeepSpeed for multi-node memory scaling

Orchestration: Slurm batch launcher for job submission

Storage: data on the shared /mnt/data

Monitoring: W&B for experiment tracking

https://github.com/dashabalashova/esm2-vhh-distributed.git

Models and DeepSpeed ZeRO

Problem: large models cannot fit on a single GPU memory-wise.

Solution:

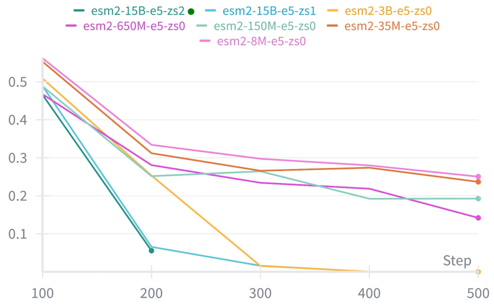
1. DeepSpeed ZeRO fine-grained memory partitioning:
   - Stage 0 – baseline (no ZeRO)
   - Stage 1 – shard optimizer states
   - Stage 2 – shard optimizer + gradients
   - Stage 3 – full parameter partitioning (params + grads + opt states)
2. 16-bit floating point precision

ESM-2 models:

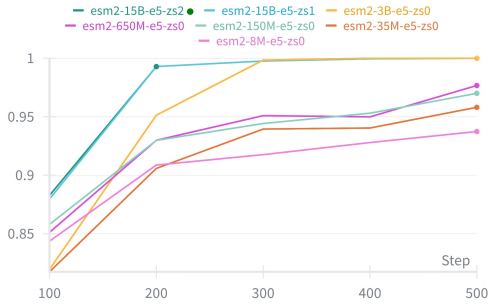
- esm2_t6_8M
- esm2_t12_35M
- esm2_t30_150M
- esm2_t33_650M
- esm2_t36_3B
- esm2_t48_15B

Model performance & Time

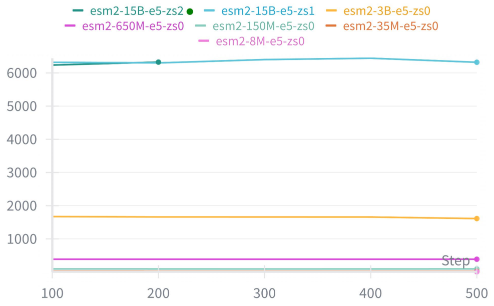
train/avg_loss



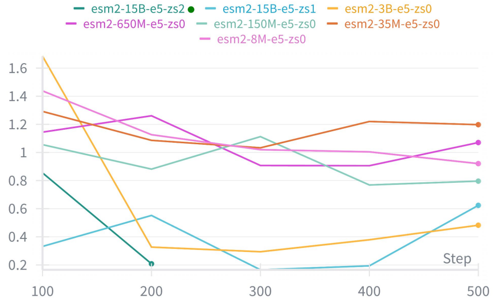
train/roc_auc



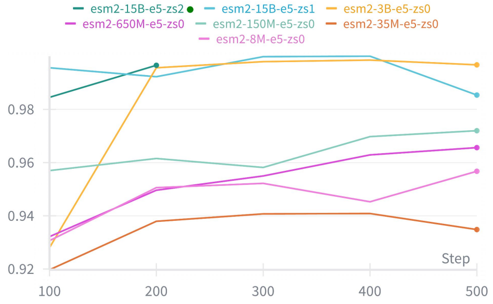
epoch_time_sec



val/avg_loss



val/roc_auc

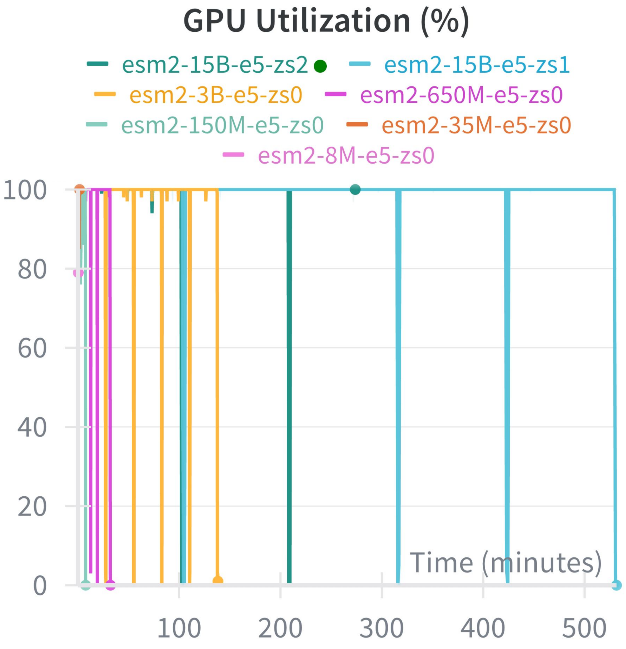


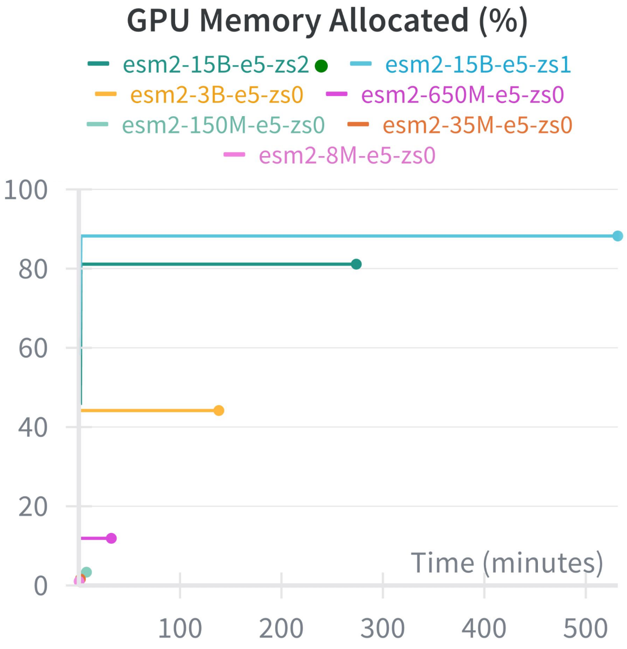
experiments:
data – 2K sequences
train/val – 0.8/0.2
15B models – sz 1/2/3 + fp16
estimated time on 2M sequences:

- 4.5 days for 650M
- 19.1 days for 3B
- 72.9 days for 15B

https://api.wandb.ai/links/dashabalashova/fh9enjaq

GPU Utilization & Memory Allocation


GPU Utilization (%)


GPU Memory Allocated (%)

https://api.wandb.ai/links/dashabalashova/fh9enjaq

Thank you for your attention!