UiT The Arctic University of Norway

BIO-AI LAB | ARCTIC LLM WORKSHOP 2023

Large Language Models

Day 2 - Session 6

Large Scale Training of LLMs and Challenges

Suyog Jadhav

suyog.s.jadhav@uit.no

28. Oct 2023

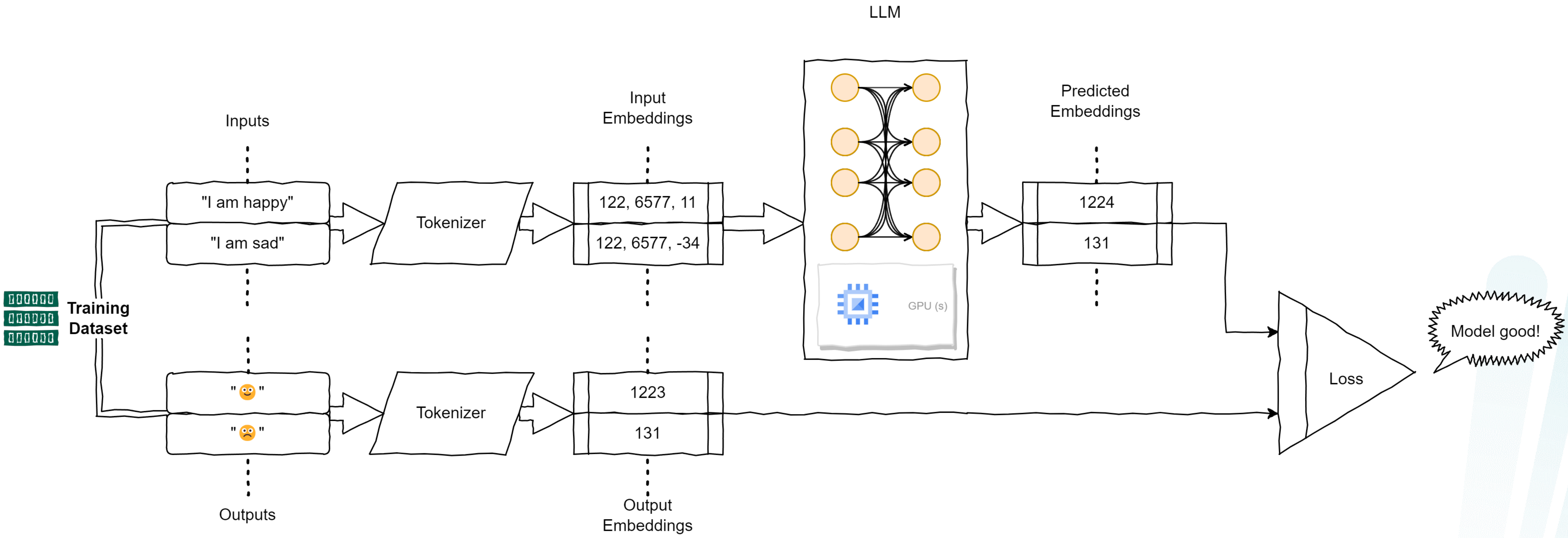
Introduction

- In this session, we will learn about all the basics you will need to know to get started training your own LLMs.

- We will also touch on some of the most prevalent challenges currently and see some ways of mitigating them.

- Finally, there will also be a programming demo at the end to help you see these things in action!

A usual LLM training workflow



Key Challenges

- Data
    - Ethical concerns
    - Bias and prejudice against groups of individuals

- Legal
    - IP violation concerns for crowdsourced training data

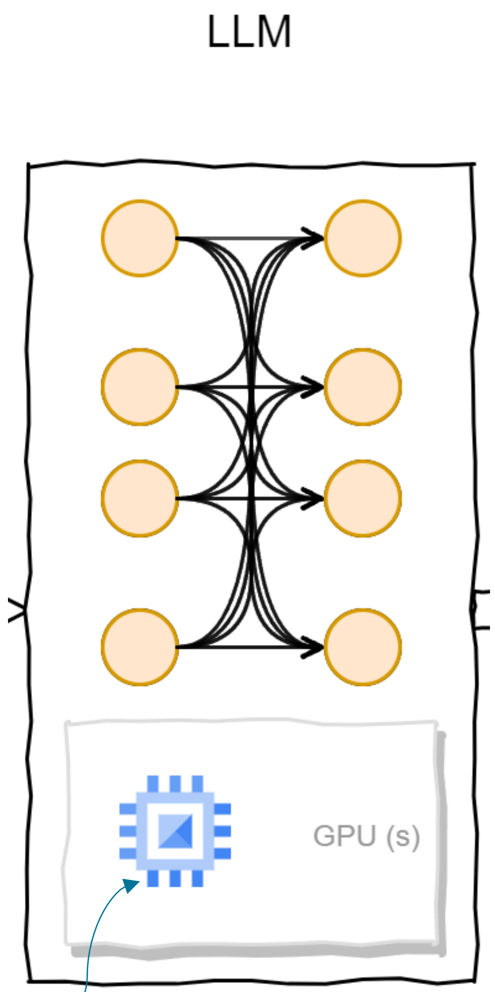
- Environmental
    - Rising concerns around large scale LLM training amounting to large carbon footprint

- Hardware
    - Most LLMs are too large to fit in today's GPUs
    - How do we overcome the memory limitation without sacrificing too much of the computational speed?

Currently, the highest amount of GPU memory available on a GPU is 80GB, on an NVIDIA A100 GPU.

The size of GPT-3

“Training can now be done on **175 billion-parameter** models **on 300 billion tokens** using **1,024 NVIDIA A100 GPUs** in *just* **24 days**–reducing time to results by 10 days, or some 250,000 hours of GPU computing, prior to these new releases.” – From Nvidia’s recent blog[1]



>1 of these is now a requirement!

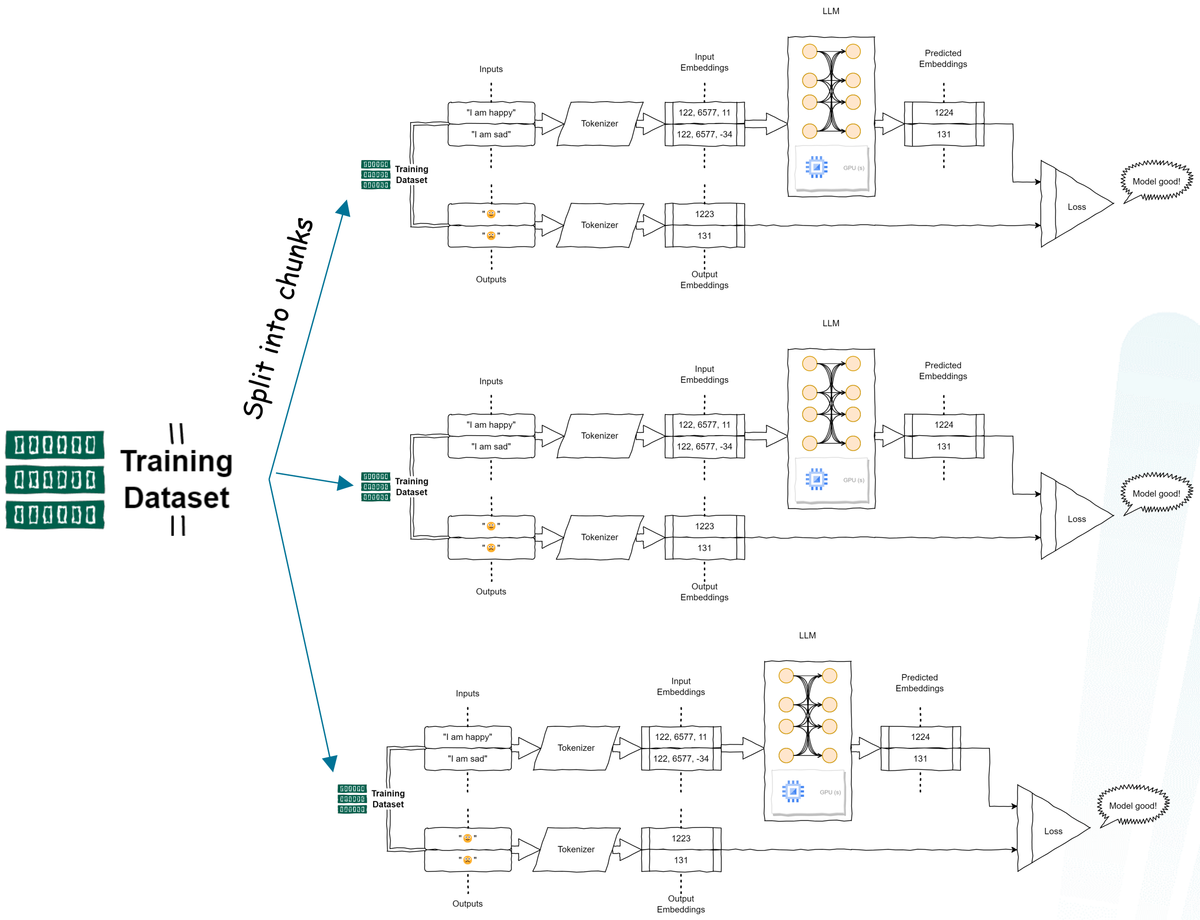
1. https://developer.nvidia.com/blog/nvidia-ai-platform-delivers-big-gains-for-large-language-models/

Solution? Parallelism

1. Data Parallelism (DP):
   Replicate the model on all the GPUs
   Split the available data across them

   + Reduces training time, as the data is being split across multiple GPUs
   + If model fits on a single GPU, this is usually the fastest of all the options
   + No modifications to the model code required

   - If the model does not fit on a single GPU, this is completely infeasible



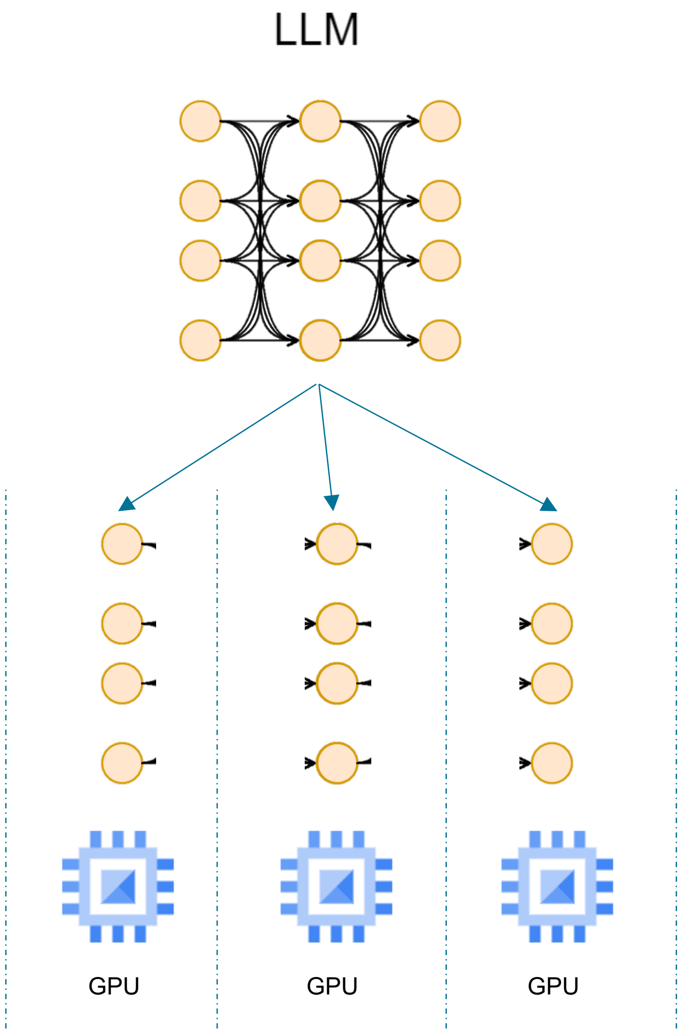
Parallelism (contd.)

2. Pipeline Parallelism (PP): Also called "vertical" parallelism, as we split the model layers by a vertical slice.

   Split the model layers across different GPUs

   + Reduces memory consumption of the model parameters on a single GPU
   + If the model does not fit on a single GPU, this is still feasible

   - Requires heavy modifications to the model code



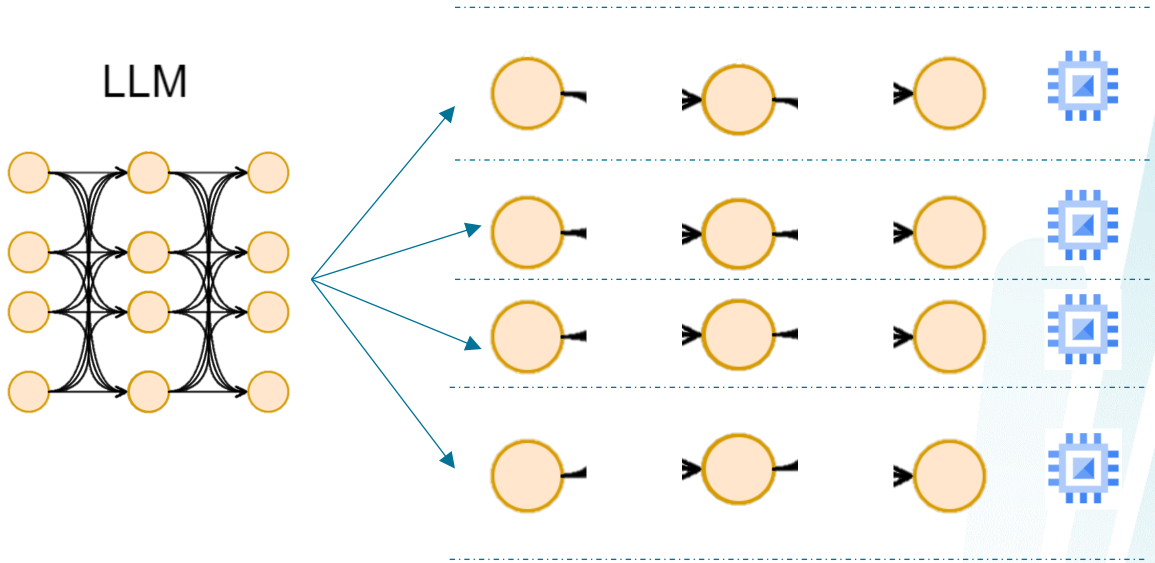
Parallelism (contd.)

3. Tensor Parallelism (TP): Also called "horizontal" parallelism, as we split the model layers by a horizontal slice.

   Splits the computation such that matrix multiplications that don't depend on each other are separated into different GPUs

   + Reduces memory consumption of the model parameters on a single GPU
   + If the model does not fit on a single GPU, this is still feasible
   + Can be made in a way that reduces aggregation steps to just once, reducing the queuing and GPU idling

   - Requires heavy modifications to the model code



Parallelism (contd.)

4. DeepSpeed ZeRO (Zero Redundancy Optimizer): Result of a recent research work from Microsoft[1].

   Splits the computation such that the model is split horizontally in each layer, and each of the GPUs fetch the required values on-demand from the GPU that has it.

   Also performs sharding of the tensors somewhat similar to TP, except the whole tensor gets reconstructed in time for a forward or backward computation, therefore the model doesn't need to be modified!

   Microsoft DeepSpeed[2] also implements many other optimizations such as layer fusion, CPU and NVMe offloading, distributed computing, etc. that effectively take the available memory space to an indefinitely large amount.

   + Reduces memory consumption of the model parameters on a single GPU
   + If the model does not fit on a single GPU, this is still feasible
   + Does not require any modifications to the model code

1. "ZeRO: Memory Optimizations Toward Training Trillion Parameter Models" https://arxiv.org/abs/1910.02054
2. https://github.com/microsoft/DeepSpeed

DeepSpeed ZeRO

The following video illustrates the workings of DeepSpeed ZeRO (stage-3) with an example training iteration done with 4 GPUs:

ZeRO 4-way data parallel training

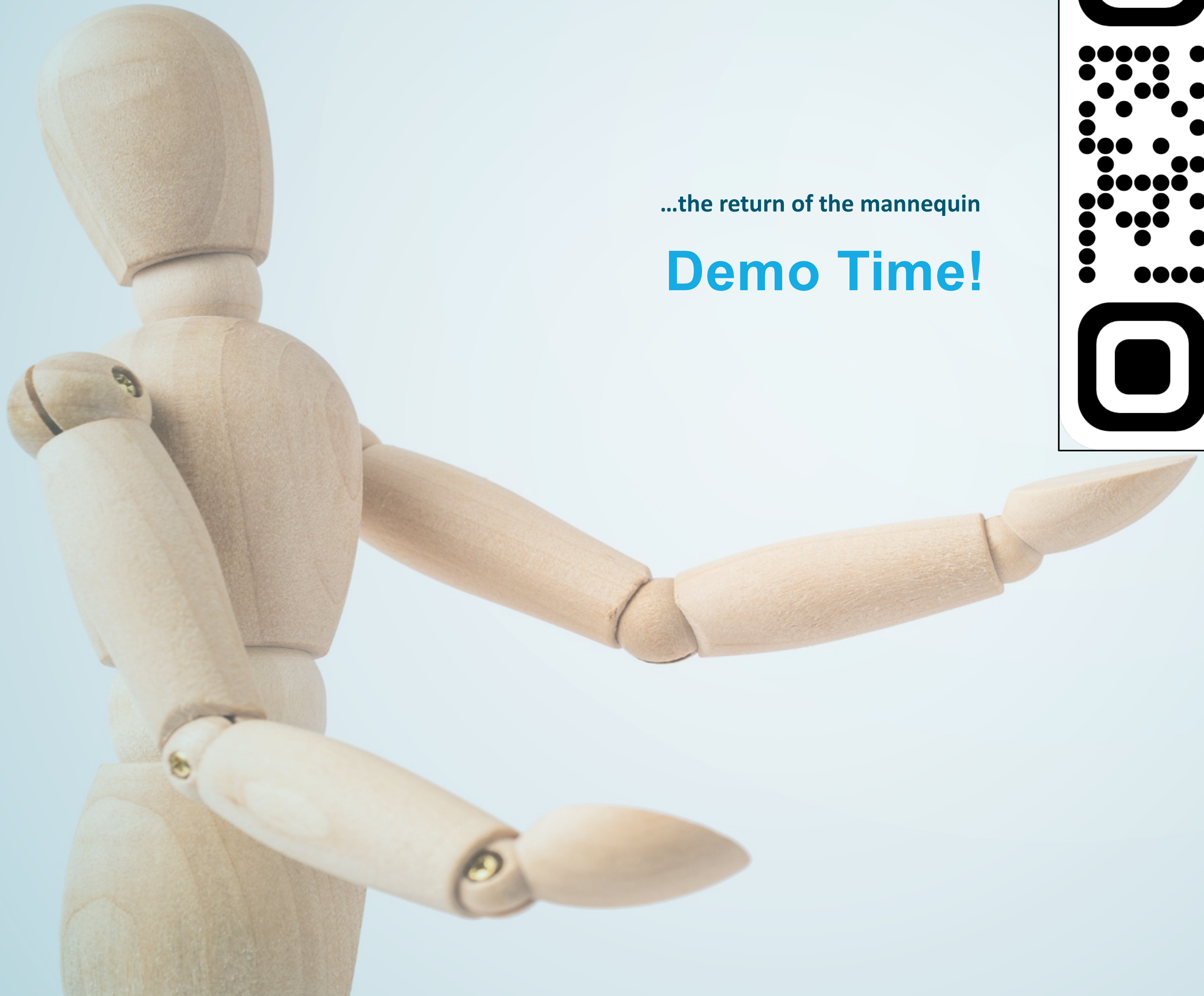
Using:

- P_{os} (Optimizer state)
- P_g (Gradient)
- P_p (Parameters)

Video from Microsoft’s blog on DeepSpeed: https://www.microsoft.com/en-us/research/blog/zero-deepspeed-new-system-optimizations-enable-training-models-with-over-100-billion-parameters/

...the return of the mannequin

Demo Time!



DEMO CODE

DEMO CODE

Question, Mark?

THE ARCTIC UNIVERSITY OF NORWAY • UiT •

https://en.uit.no/enhet/ifi

Thank You

Arctic LLM Workshop 2023
Dept. of Computer Science

bioai

www.bioailab.org