

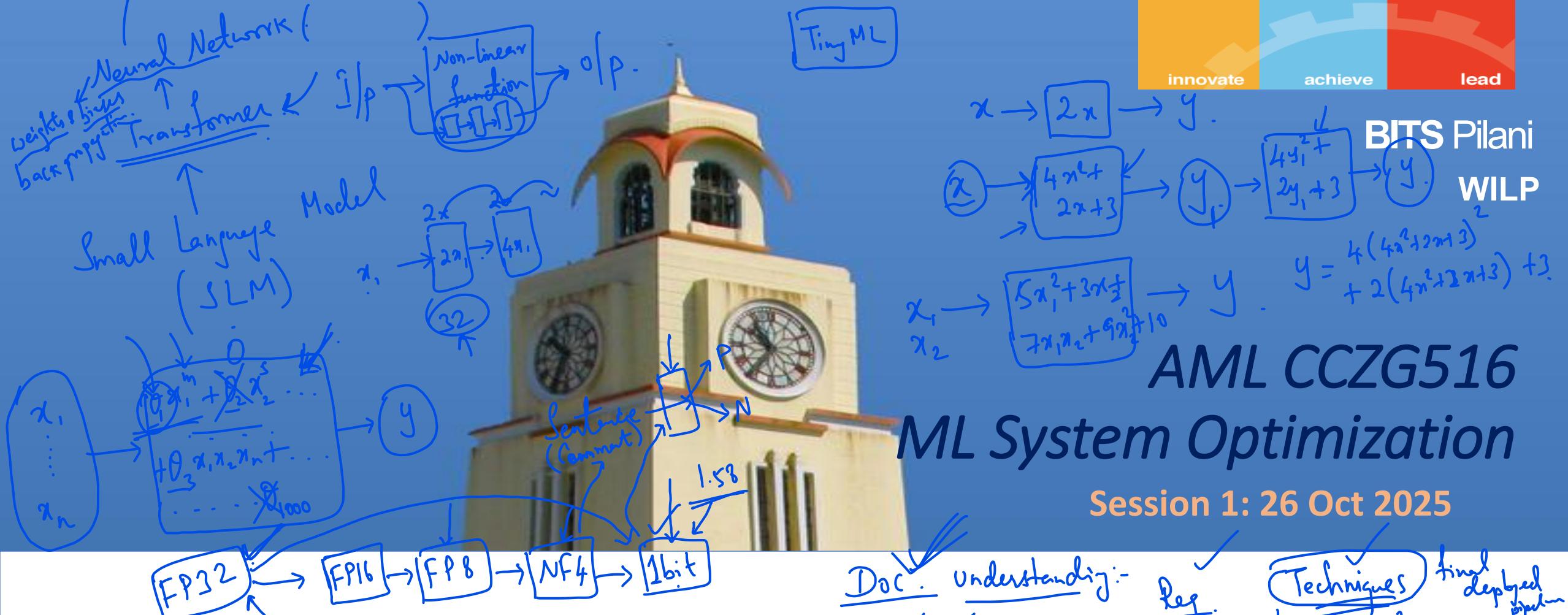


**BITS Pilani**  
**WILP**

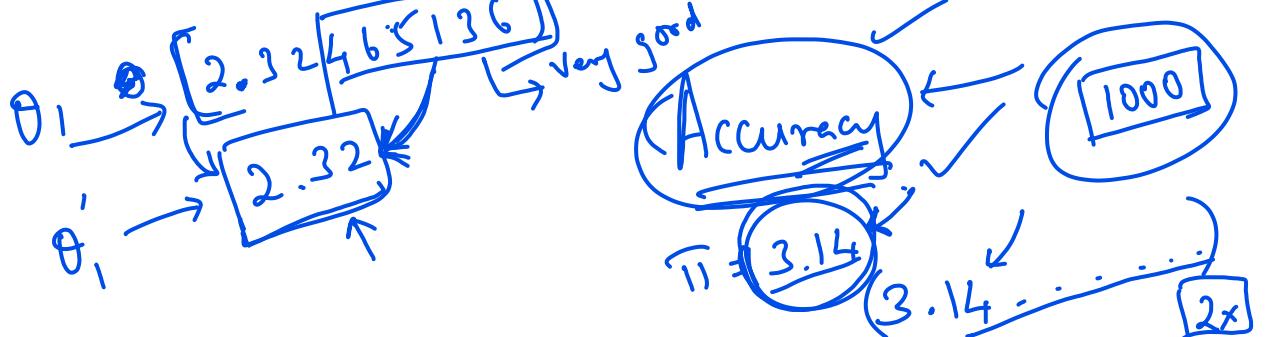
*AMLCCZG516*  
**ML System Optimization**

Murali Parameswaran



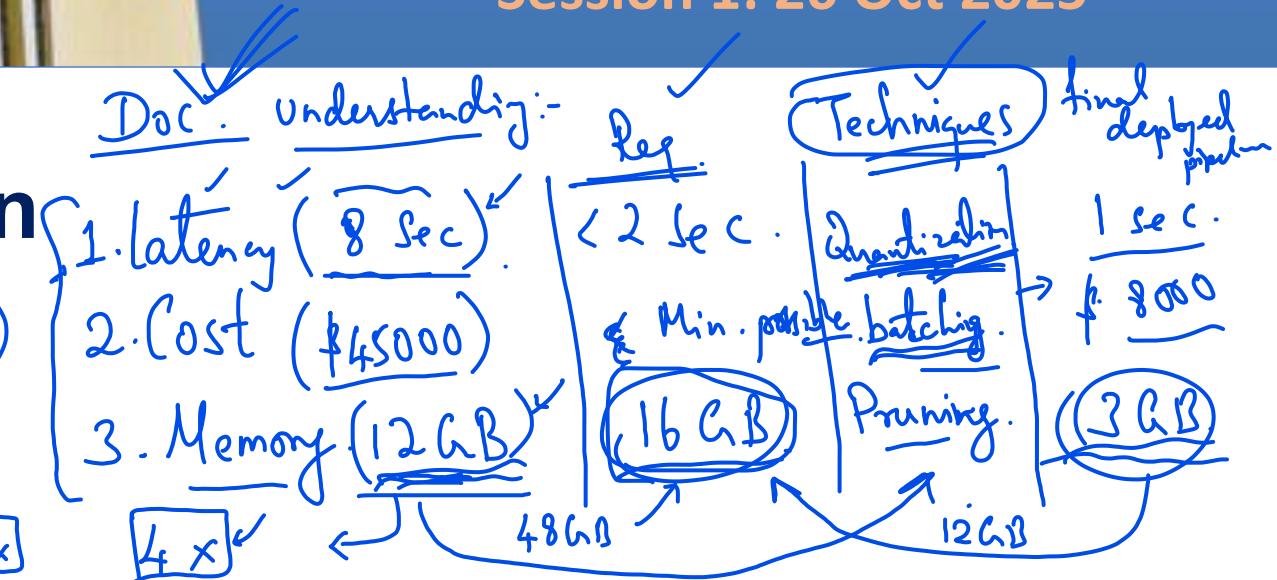


## Orientation: Course Introduction



## ML System Optimization

Session 1: 26 Oct 2025

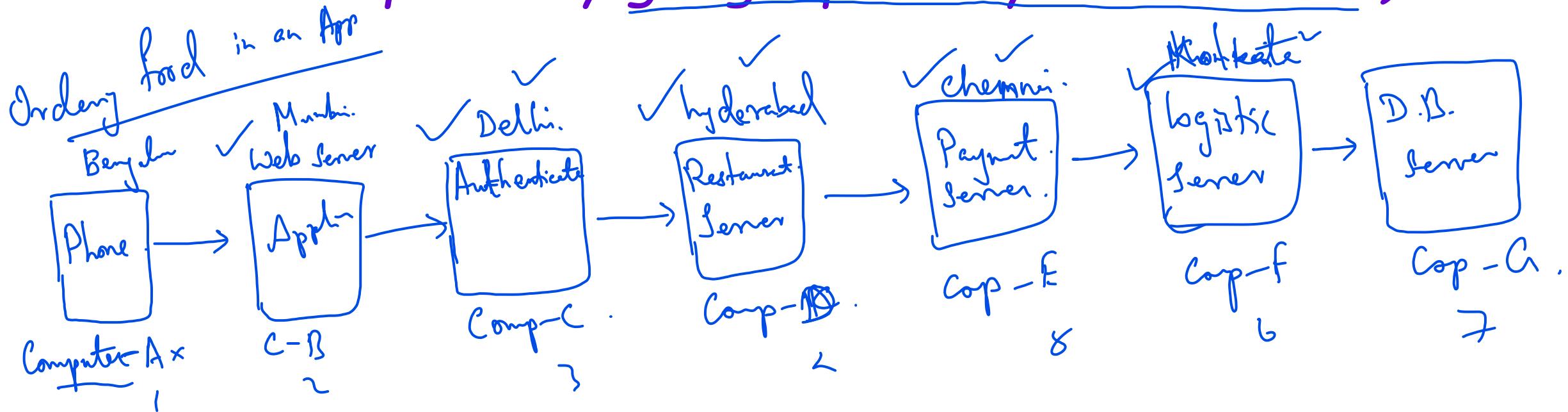


$$y = \frac{4(4x^2 + 2x + 3)}{2(4x^2 + 2x + 3)} + 3$$

AML CCZG516

## Distributed Computing

Running programs on multiple computers (in a network - possibly geographically distributed)



# *Distributed Computing*

✓ Infrastructure (Hardware & Systems Software)  
for  
Applications (Algorithms, Software Solutions)

# Content & Pedagogy

- Focus on principles, concepts, and design
  - Pragmatics and Implementation to be learnt by doing - enabled by Assignments and Project.
- Lecture Sessions are expected to be interactive:
  - students are expected to raise questions and
  - the instructor will ask questions (which the students are expected to answer)

## ✓ Evaluation

- Mid-term test and final exam - centrally scheduled by BITS
  - A Total of 50% weight
- ✓ Assignment(s) and a Project
  - $(10+10+30 =) 50\%$  ✓
- Recommended Programming Environment:
  - Java (preferably) or Python
  - C/C++/Rust
- Note: No prescribed textbooks for this course.



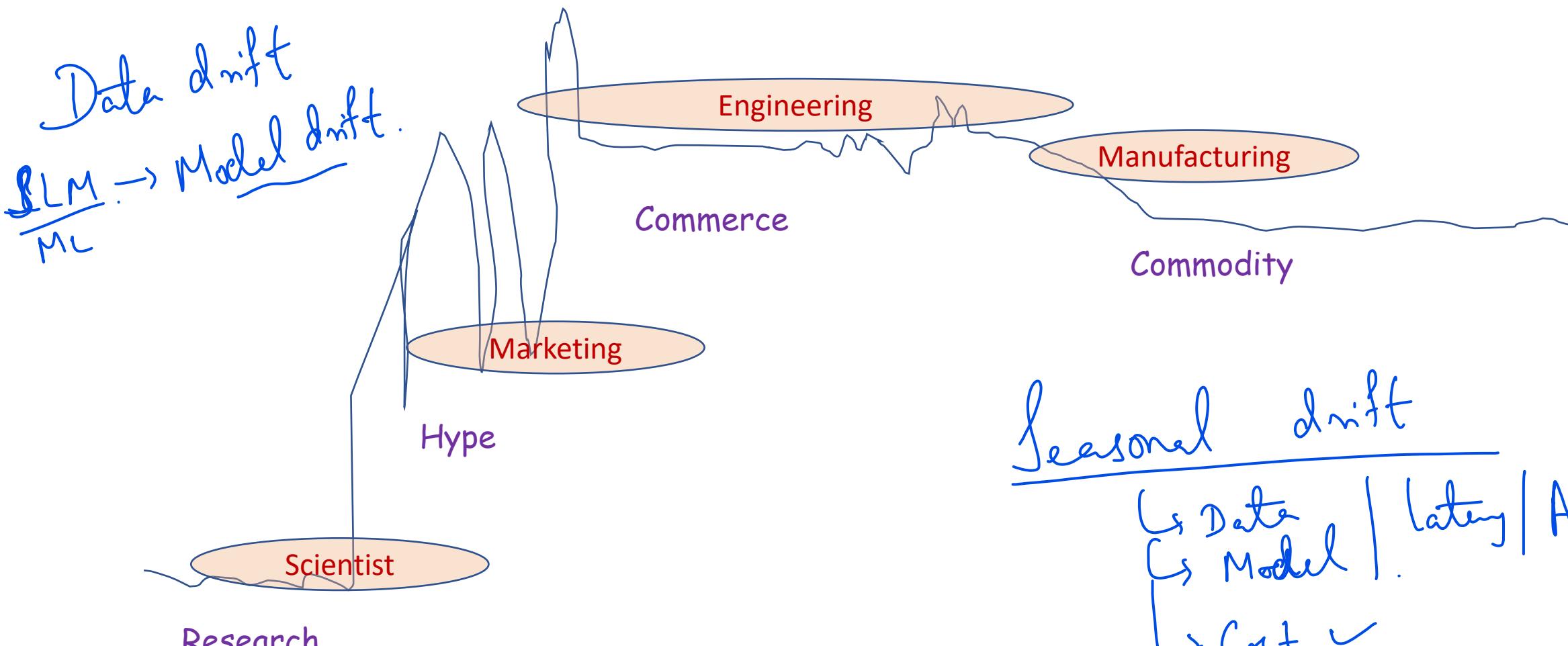
**BITS Pilani**  
**WILP**



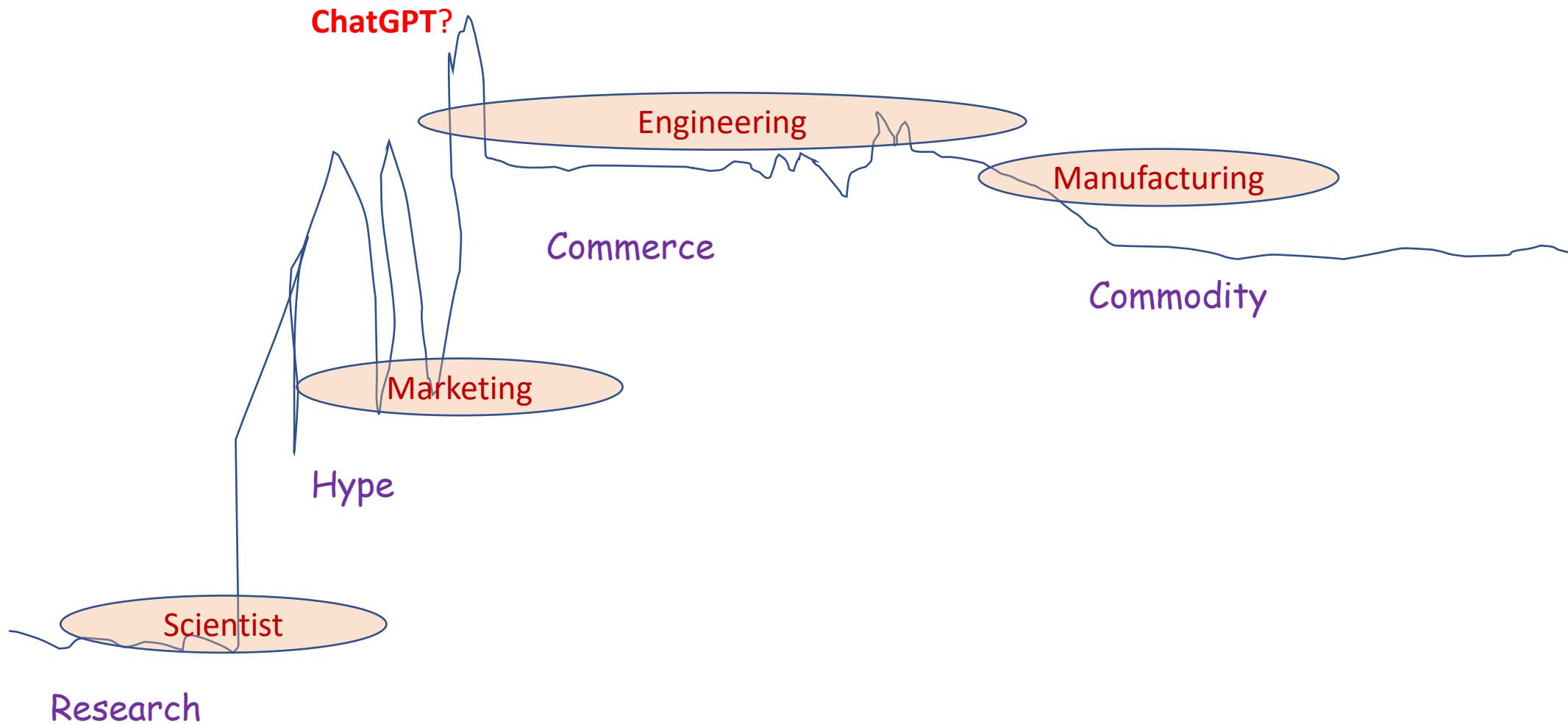
*AIML CLZG516  
ML System Optimization  
Session 1*

## **Course Introduction**

# Lifecycle of New Technologies



# Lifecycle of AI/ML - Where are we ?



# Machine Learning - Enterprise Practice

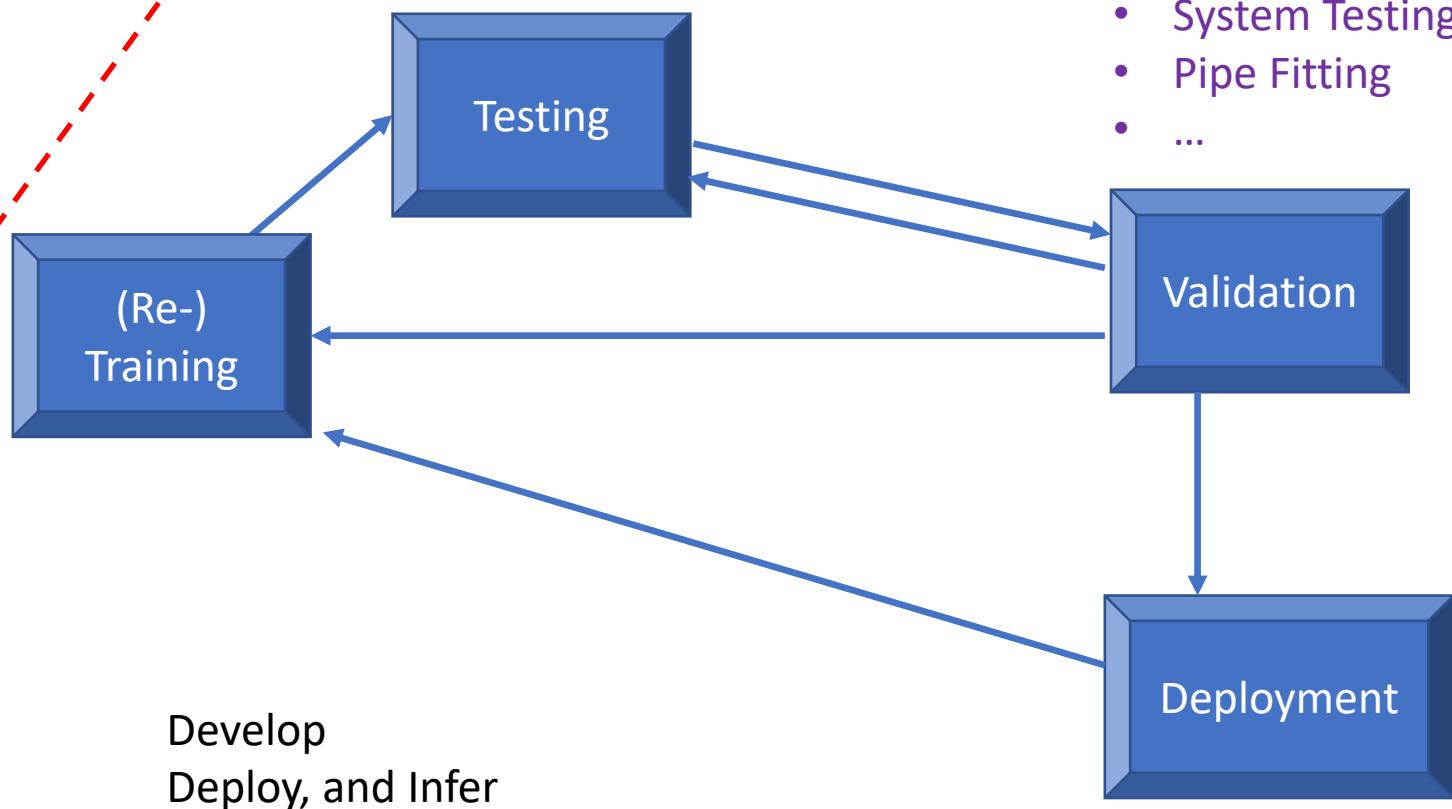
- AI and Machine Learning is becoming central to organizations:
  - No longer a one-off activity
  - Multiple problems / perspectives addressed through ML
  - Multiple ML solutions deployed
- ML is becoming a continual activity:
  - Data change; Context changes
    - Drift in the solution
  - Problems change; Requirements change;
    - New model(s) required
  - World changes; Expectations change
    - Performance and Standardization critical ==>
      - Packaging vs. Pricing

# Operationalizing AI/ML

- Class-room View :
  - Development
    - Model Building / Training
  - Deployment
    - Inference

- Enterprise View:

- Compliance
- System Testing
- Pipe Fitting
- ...

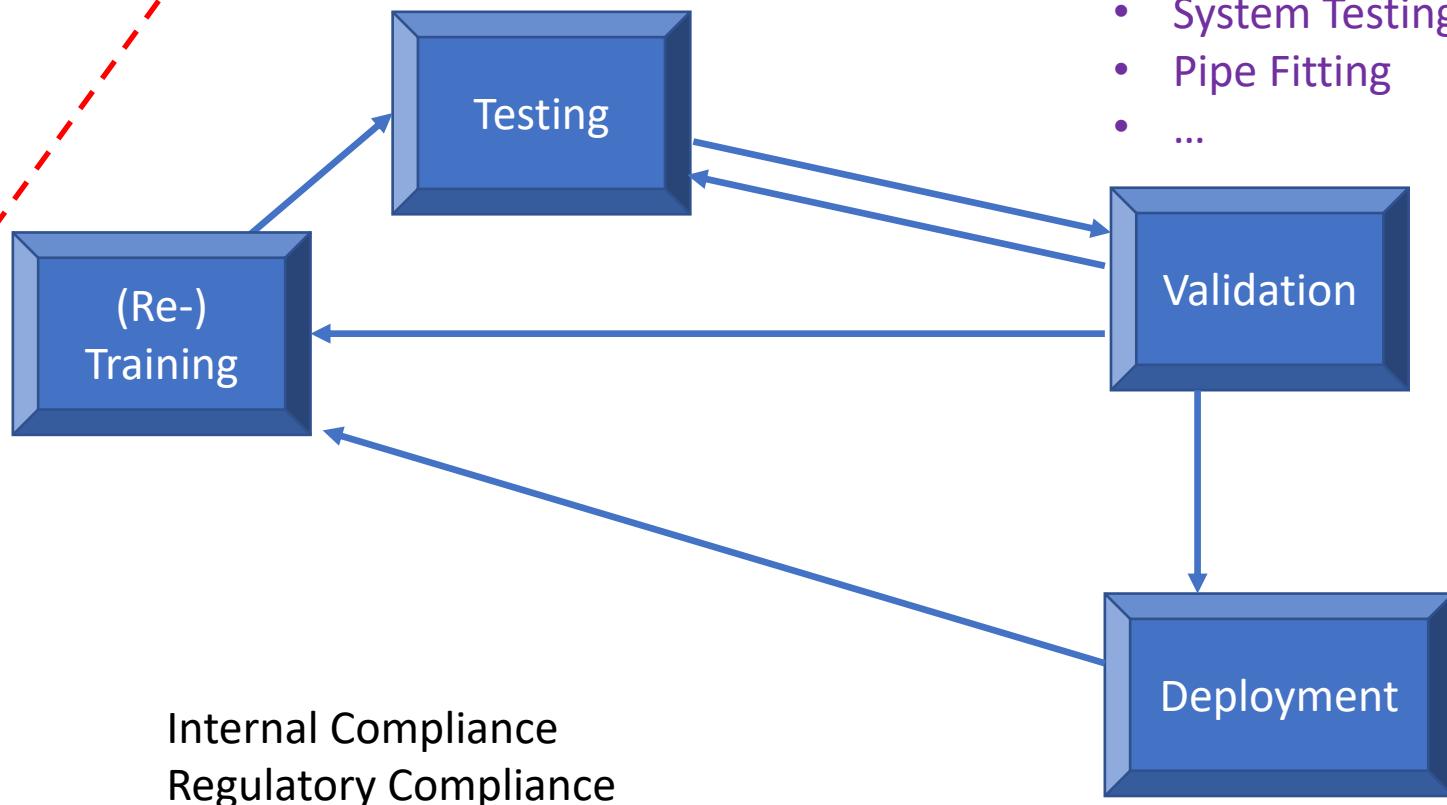


# Operationalizing AI/ML

- Class-room View :
  - Development
    - Model Building / Training
  - Deployment
    - Inference

- Enterprise View:

- Compliance
- System Testing
- Pipe Fitting
- ...

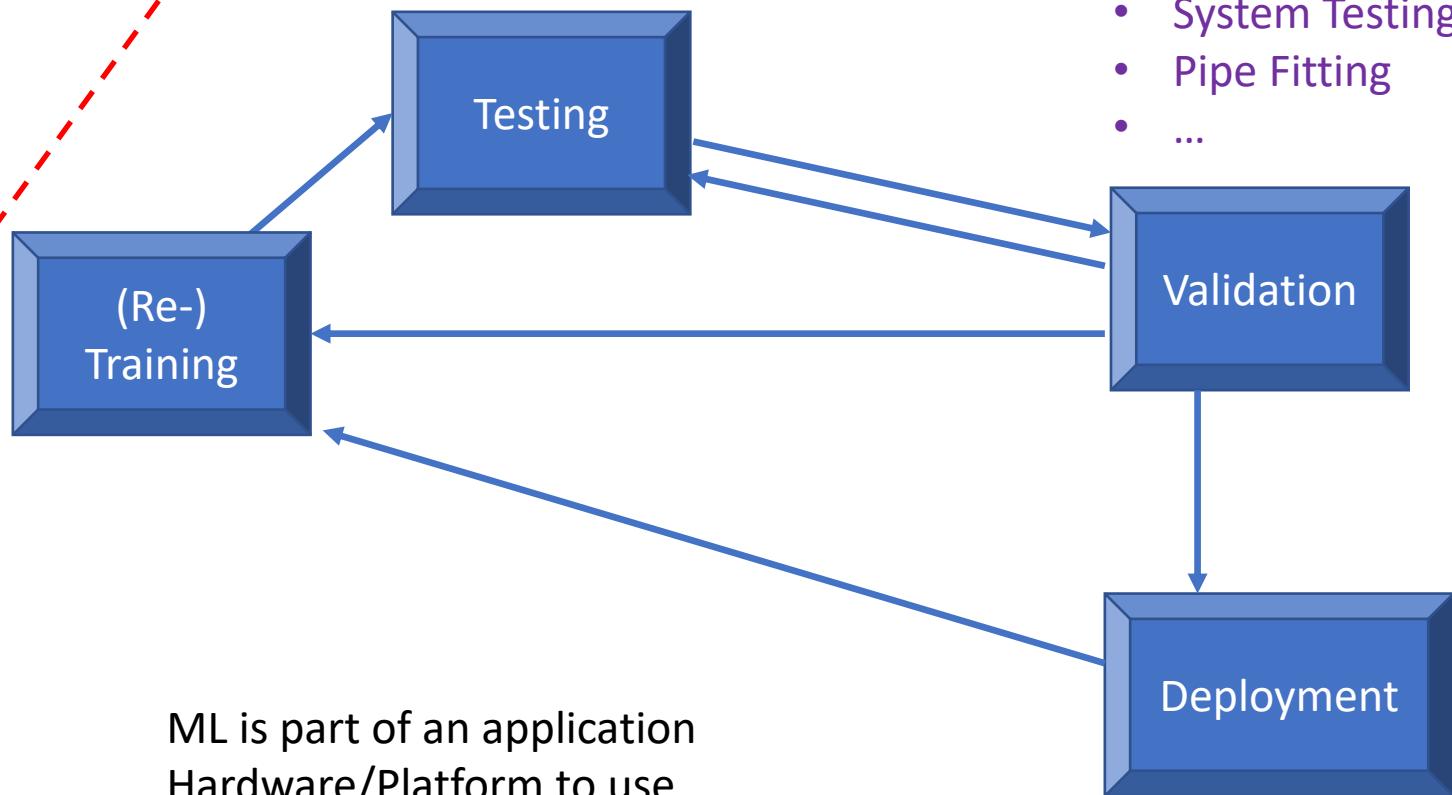


# Operationalizing AI/ML

- Class-room View :
  - Development
    - Model Building / Training
  - Deployment
    - Inference

- Enterprise View:

- Compliance
- System Testing
- Pipe Fitting
- ...

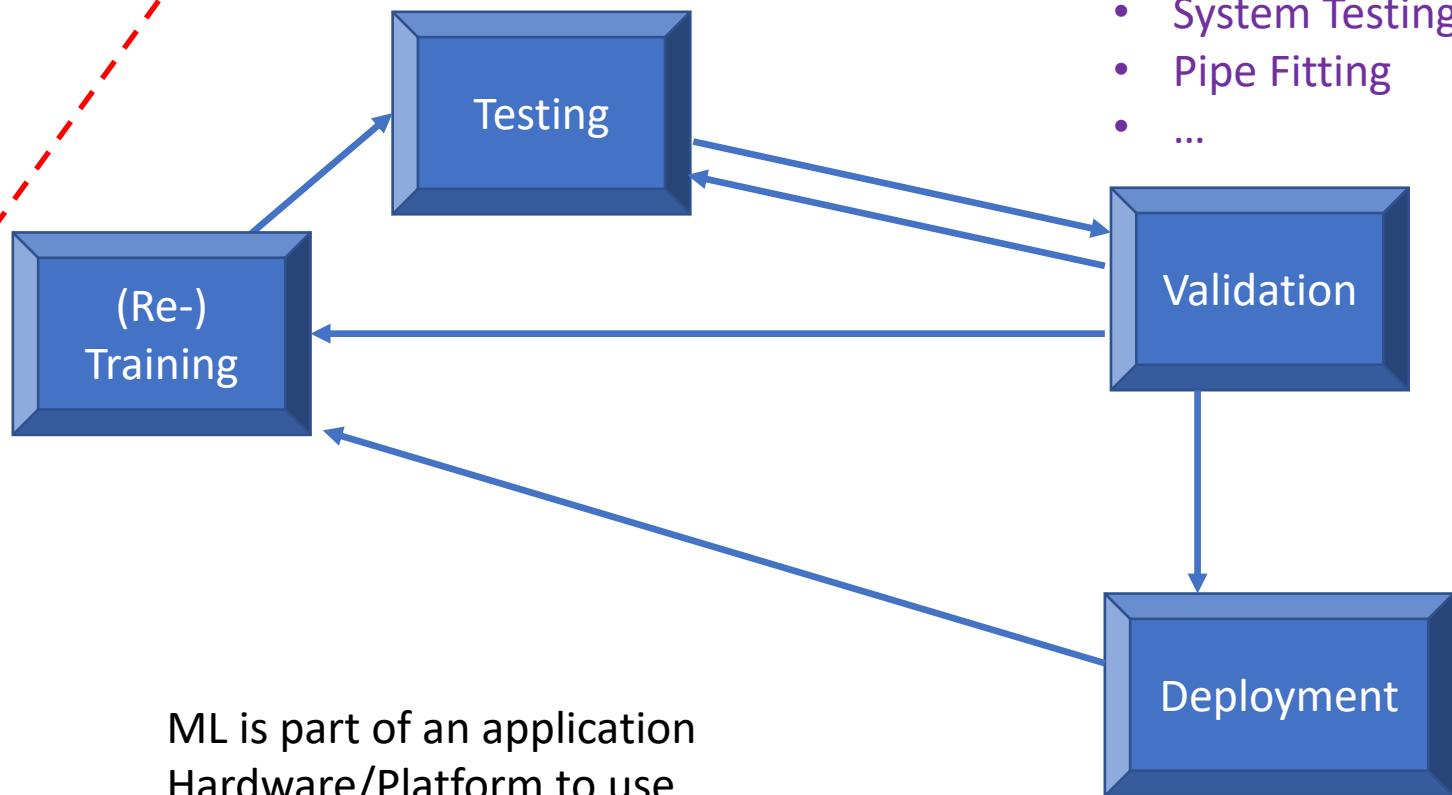


# Operationalizing AI/ML

- Class-room View :
  - Development
    - Model Building / Training
  - Deployment
    - Inference

- Enterprise View:

- Compliance
- System Testing
- Pipe Fitting
- ...



Will our model fit in the pipeline?

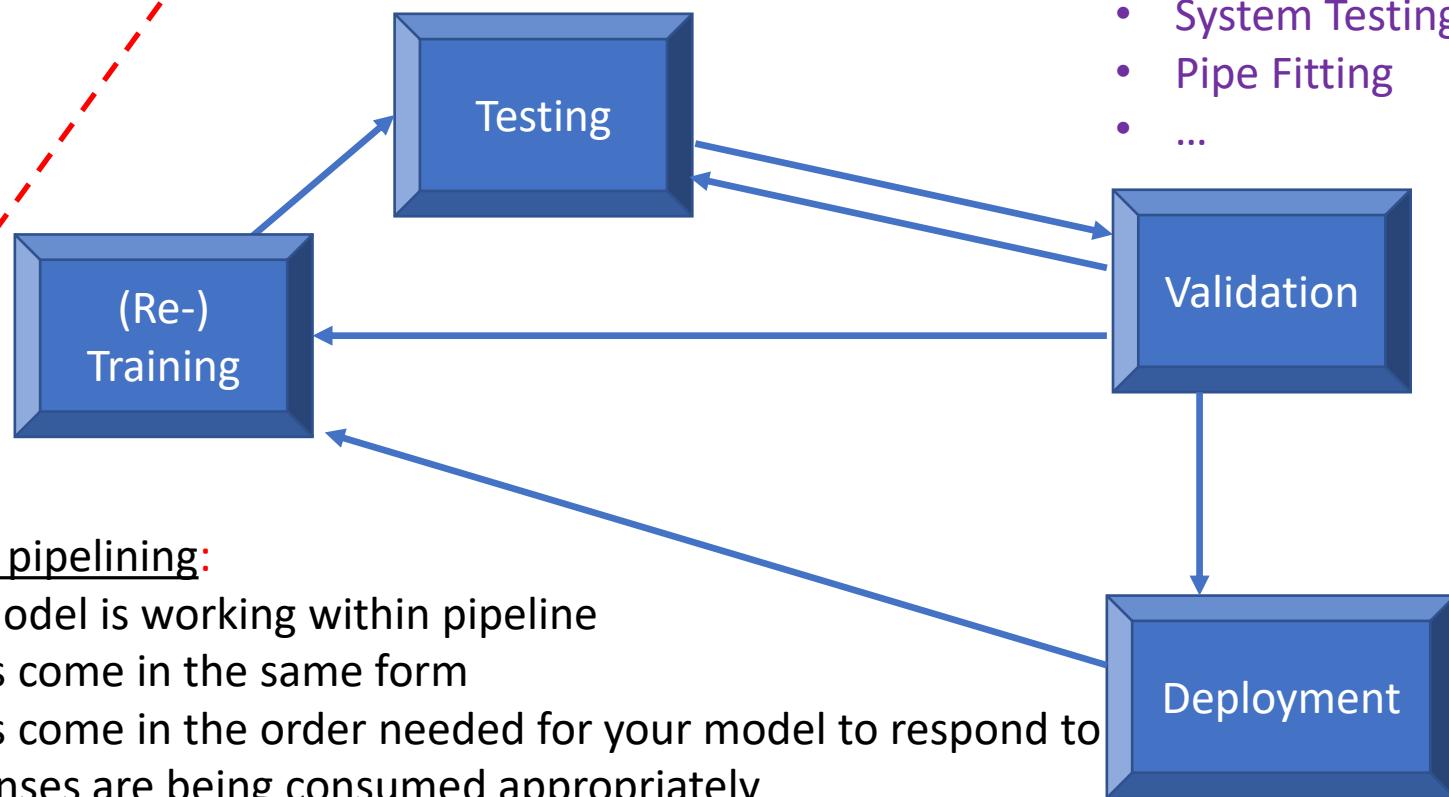
# Operationalizing AI/ML

- Class-room View :

- Development
  - Model Building / Training
- Deployment
  - Inference

- Enterprise View:

- Compliance
- System Testing
- Pipe Fitting
- ...



Software pipelining:

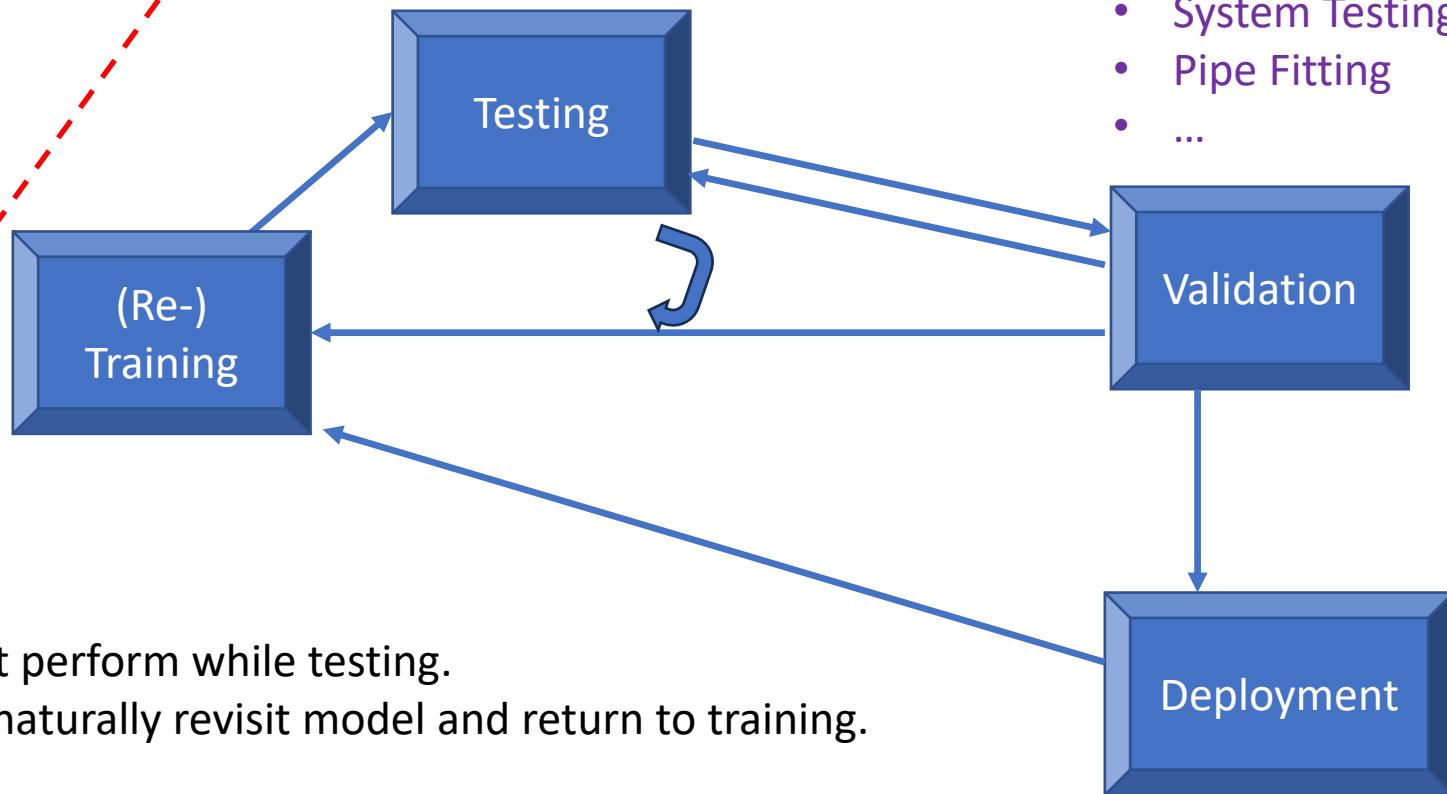
- Test whether model is working within pipeline
- Whether inputs come in the same form
- Whether inputs come in the order needed for your model to respond to
- Whether responses are being consumed appropriately
- Whether responses are to be consumed one at a time, or a sequence of responses have to be consumed

# Operationalizing AI/ML

- Class-room View :
  - Development
    - Model Building / Training
  - Deployment
    - Inference

- Enterprise View:

- Compliance
- System Testing
- Pipe Fitting
- ...



May need to return to training after validation/testing.

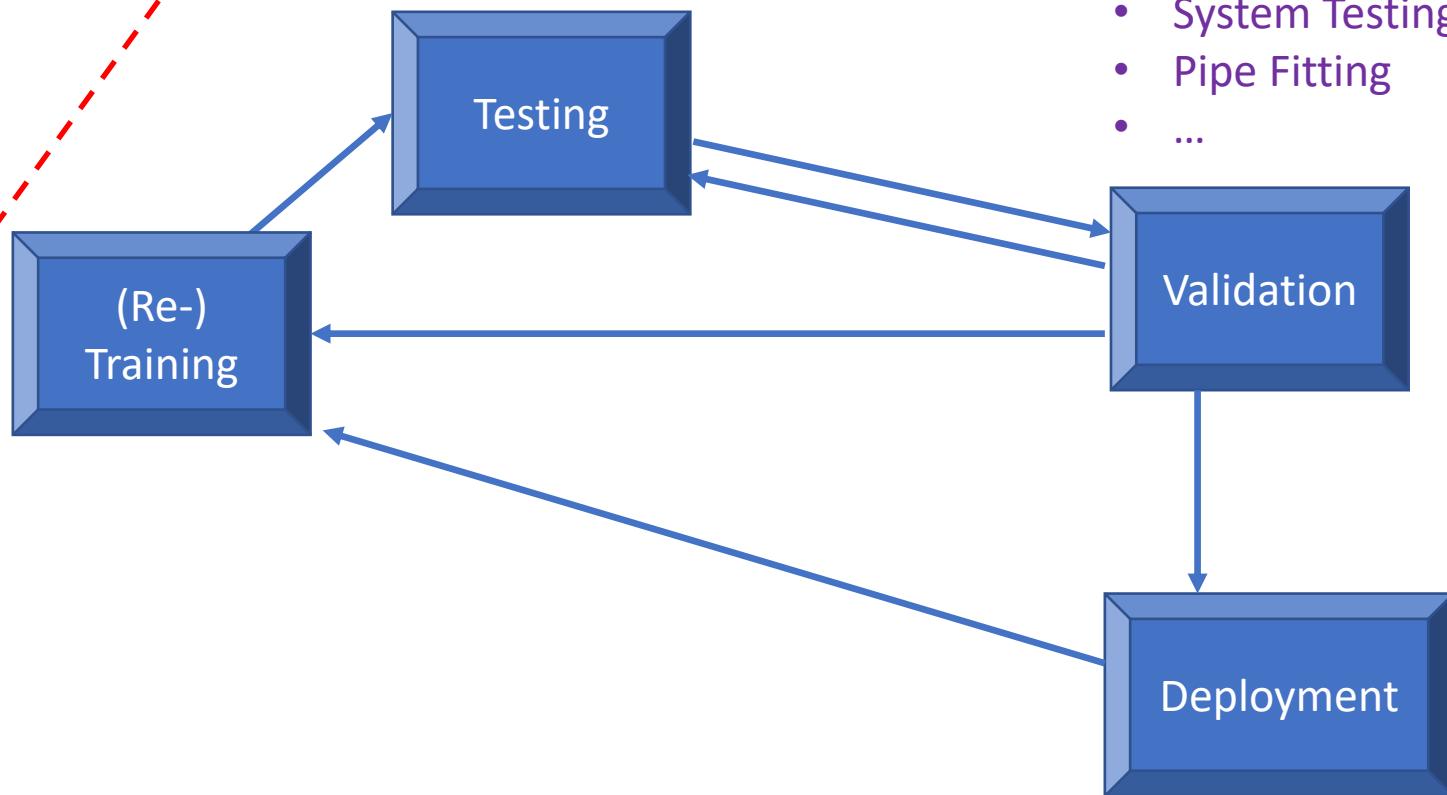
# Operationalizing AI/ML

Can we stop after deployment?

- Class-room View :
  - Development
    - Model Building / Training
  - Deployment
    - Inference

- Enterprise View:

- Compliance
- System Testing
- Pipe Fitting
- ...



# Machine Learning - Enterprise Practice-Recap

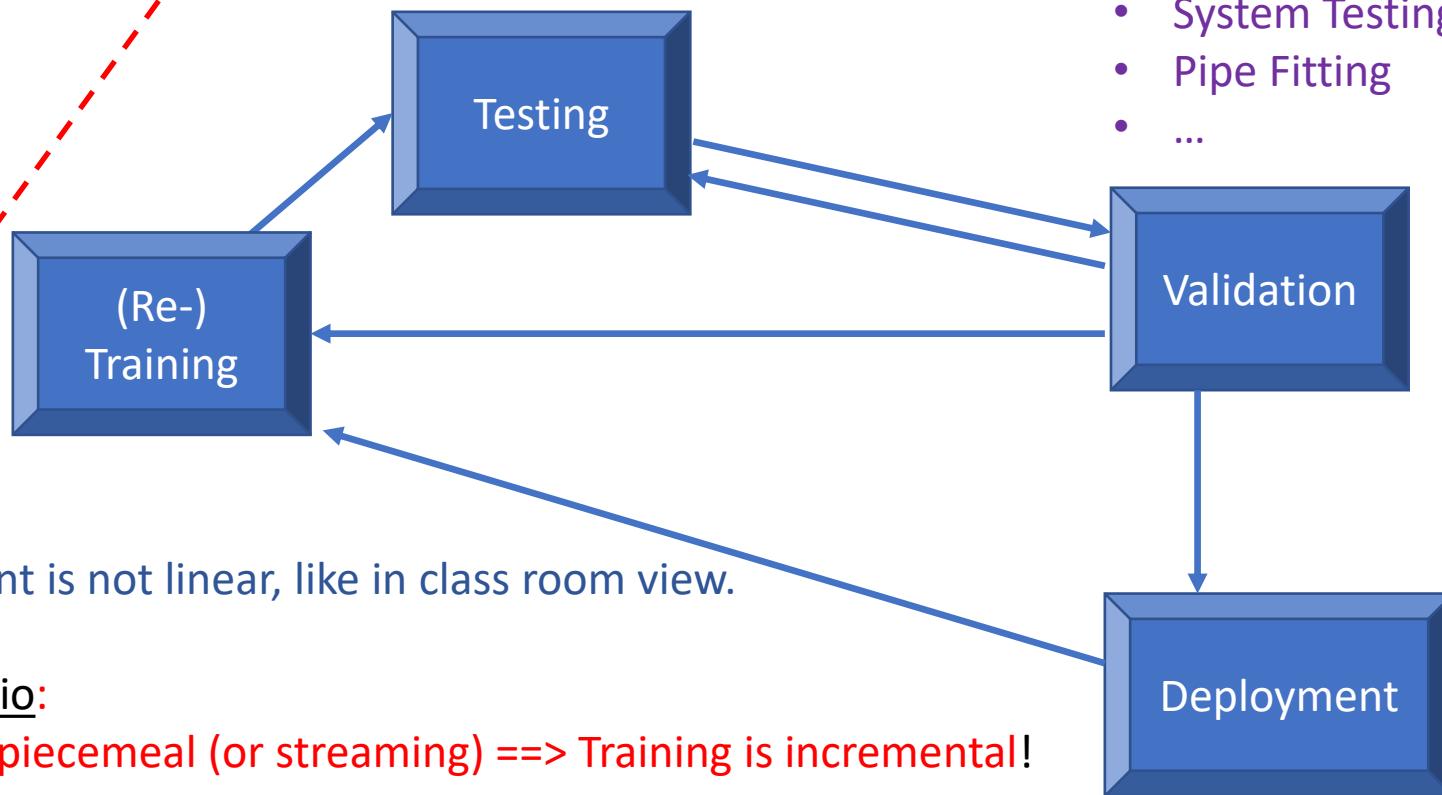
- AI and Machine Learning is becoming central to organizations:
  - No longer a one-off activity
  - Multiple problems / perspectives addressed through ML
  - Multiple ML solutions deployed
- ML is becoming a continual activity:
  - Data change; Context changes
    - Drift in the solution
  - Problems change; Requirements change;
    - New model(s) required
  - World changes; Expectations change
    - Performance and Standardization critical ==>
      - Packaging vs. Pricing
        - Depends on competition
        - Cost for model inference

# Operationalizing AI/ML

- Class-room View :
  - Development
    - Model Building / Training
  - Deployment
    - Inference

- Enterprise View:

- Compliance
- System Testing
- Pipe Fitting
- ...



How is performance delivered?

Performance metrics?

# *Operationalizing AI / ML: Cost*

Focus of this course

- Cost:
  - Time and Resources during Training vs Inference
- During Training:
  - Running Time of an algorithm:
    - E.g. k-means is an  $O(N^*N)$  algorithm given N data points
    - E.g. SVM has a time complexity between  $O(d^*N^2)$  and  $(d^*N^3)$  where
      - d is the number of dimensions (of the data points) and
      - N is the number of data points

# Cost during Training

- Example
  - E.g. SVM has a time complexity between  $O(d*N^2)$  and  $(d*N^3)$  where
    - d is the number of dimensions (of the data points) and
    - N is the number of data points
  - For a large dataset N, say,  $N = 10^9$  and  $d=5$  this could be costly:
    - Assuming 2 simple arithmetic operations per data point:
    - this amounts to at least  $10^{19}$  ( $=5*2*10^9*10^9$ ) operations
  - Given a 2.5 GHz processor, i.e. 0.4ns clock cycle
  - and 1 CPI (i.e. cycles per instruction), a measure of processor throughput
    - [simplistic but close to reality!]
  - $10^{19}$  operations will take close to 5.3 years
- Reducing running time during training is a big focus in this course!

# Reducing running time

- Typical methods:
  - Parallelize or distribute computation:
    - Multi-threaded programming on multi-core processors
    - Massively multi-threaded programming on many-core GPGPUs
    - Distributed Programming on Scale-out Clusters of CPUs or GPUs
  - Hand-tuning or compiler-performed code optimization
    - Rewritten for parallelism or generated by compilers
  - Process = Program + Address Space (at run time)
    - Threads share address space:
      - Each thread gets its own call stack
      - Heap and global area are shared by all threads
      - Threads run on a shared memory model (e.g., multi-core, many-core processors)
    - Distributed programming is on Distributed memory ie. Memory of multiple computers (Processor+memory+disk+OS)

# Cost during training

- Megatron-Turing NLG:
  - 530 billion parameters
- Microsoft and Nvidia claim to have used hundreds of DGX A100 servers
  - Each server costs ~200,000 \$
    - Add the networking cost, the infrastructure cost is ~100M\$
  - Each server consumes 6.5kW of power
    - Add a comparable cooling cost!
- We will NOT do much about power consumption in this course!
  - But we will look at reducing model size as an important aspect!

# Sizes of NLP models over the years

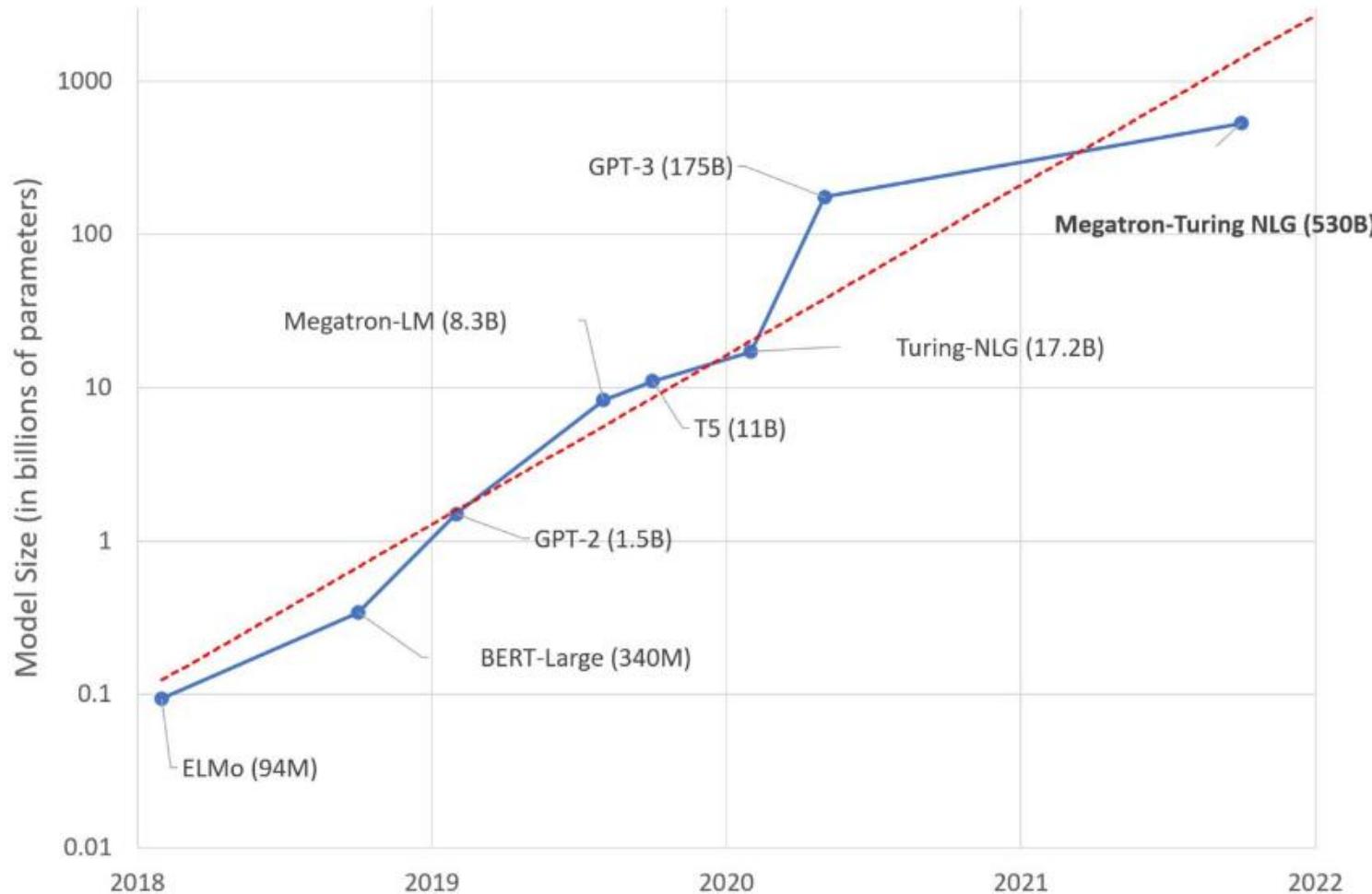


Figure 1. Trend of sizes of state-of-the-art NLP models over time

# Model Size

- LLMs (Large Language Models) like GPT-4 and Bard are notoriously large.
  - But there are systematic approaches to reduce model size
    - Without compromising the accuracy too much.
  - We will look at model compression in this course!

# Cost during Deployment

- When a model is deployed:
  - Requests come in and the model responds with inferences
    - E.g. if your model is a classifier:
      - For a new input  $x$ ,
      - the response is  $C(x)$  such that  $x \in C(x)$
- Performance Parameters for this phase:
  - Throughput:
    - Number of inferences over a unit of time
  - Response Time: Time take to serve one inference
    - Consider the classifier example with a (data) cluster example!
    - Will there be a difference in response time?

# Deployment Range

- The model (that has been trained) or an application using the model could be deployed on a variety of platforms:
  - A server (or a workstation)
    - What if the model is large?
  - The Cloud
    - The cloud can provision large infrastructure to host a large model:
      - Increase the number of servers hosting and accepting requests thereby improving throughput and response time!
        - But there is always delay
          - i.e., network latency in reaching a remote server or a server on the cloud (and getting the response back)
    - A mobile phone:
      - best end-user response time but cannot host large models.

# Sequential vs. Batch

- BATCH (Assumption): Requests are collected together and sent
  - Responses are collected together and sent
- Ans.
- Part (A) If the model server is parallel multiple threads or processes could respond in parallel thereby improving response time and throughput.
- Part (B) [Always] Messaging/Communication cost may be reduced:
  - Communication cost = setup-cost + transmission cost
    - set-up cost is fixed per message
    - Transmission cost is proportional to the length of the message

# Content & Pedagogy

- Focus on systems, programming techniques, and analysis
  - Pragmatics and Implementation to be learnt by doing - enabled by Assignments and Project.
- Lecture Sessions are expected to be interactive:
  - students are expected to raise questions and
  - the instructor will ask questions (which the students are expected to answer)

# Assignment and Project

- They are meant for learning
  - Expected:
    - One complex-end-to-end piece of optimization completed
    - One cutting-edge optimization technique learnt
  - Team-work with identifiable and quantifiable individual contributions
    - Evaluation both at team level and individual level