MOTIVATION

Software 2.0, also known as machine-learned software, is now ubiquitous and typically, the ML component is in the critical path of the system.

As such, the performance and reliability of the system depend on the accuracy and performance of the ML model.

These systems usually operate in non-static environments, which are prone to changing unpredictably which affects the performance and accuracy of the ML model.

PROBLEM

• How can we detect when ML-based systems are performing poorly and need to be adapted?

• What can we do to improve the ML model in run-time? What tactics do we have available?

• How do we choose the tactic to execute? What are the costs and benefits of each tactic?

GOAL

Build a repertoire of tactics to repair ML models that require runtime adaptation

ADAPTATION TACTIC EXAMPLES

• Re-train ML model
• Transfer learning
• Model patching
• Forgetting samples