allegro AI

Case Study

AI Workflow Automation with - and for - Surgical Precision

Faced with building their own solution and hiring Dev-Ops staff, Theator found a solution that both saved money and drove productivity.

Theator: Client Overview

All surgeries, no matter how long or complex, are defined by a handful of critical moments. Moments where vital, split-second decisions determine whether surgery succeeds, or doesn't. Where knowing the patient and procedure from all angles is the difference between preventing complications...or creating them. The team at Theator, an innovative up and coming company, is leveraging computer vision and machine learning to introduce Surgical Intelligence, a platform that puts defining intraoperative moments in the hands of surgeons so they can continuously perfect their craft. With Surgical Intelligence, surgeons have the technological edge needed to turn precious, data-enriched moments into smarter decisions, sharper skills, and better outcomes.

The Challenge:

Hardware Automation and Visibility

At the crossroads of computer vision, surgery, and artificial intelligence, Theator's data science team requires extensive computing resources to build, test, and refine their models to incorporate every permutation of each type of surgery they aim to improve. As is the norm in AI development, the data scientists at Theator first spend enormous amounts of time creating the initial experiments, then apply the models to massive, complex visual data sets, and then leave them to run while they're working on modified versions of that experiment.

After careful analysis, Theator decided that for their needs, it didn't make sense to purchase a large bank of expensive GPU machines, especially when, aside from peak usage, many would not be in use. "Instead of managing our own hardware for software development", explains Dotan Asselman, CTO and Co-founder of Theator, "we use Cloud computing so we can spin machines up and down as needed. The downside, of course, is that the costs associated with the time, storage and bandwidth are enormous when we aren't careful to keep an eye on what is running and when". To management, allocating dedicated DevOps staff for this seemed wasteful. However they knew they needed a solution to empower their data science team to control and streamline the process and to keep machines running only when they had to be. The problem, of course, was the time and distraction involved in doing so.

The Solution:

As they learned about Allegro Trains, the team believed it had their answer. Trains is an open-source platform that helps data science and data engineering teams optimize their AI development with a number of tools that yield immediate improvements across four core areas: productivity, collaboration, resource utilization and data management. In less than a week, the Theator

team had configured the Trains ML-Ops module, and fully integrated it into their environment. Trains now automatically spun machines up and down with cost-saving efficiency, based on actual demand triggered by the Trains Agent, a component of Allegro Trains.

> Using the Allegro AI platform, machines can be automatically spun up and down on the fly, prompted by actual demand, without any middle-man intervention by DevOps.

Allegro Trains provides an interface to monitor workers and queues that can, on one hand, migrate experiments on demand, and on the other hand, actually execute the experiment...all without requiring the developer to set up the machine, install packages, and re-write code. As these tasks are handled automatically; Developers no longer need to first access a given machine and check its "vital signs" to determine if it has the capacity to manage another experiment.

"A major benefit for us," explained Asselman, "is that Trains-agent includes its own machine status reporting engine to monitor GPU, CPU, VRAM, RAM allocation, and network IO. As such, we can now trace the impact of specific code, in a given experiment, on any of these resources. This is a critical data point we often need to discover; when training crashes, we need to know why it happened so we can get back to work. And if my code isn't maximizing the GPU's capacity, I'm wasting time and money."

To achieve this level of transparency through other means, Theator had to either build their own solution, or purchase and manage an external solution. "It feels like Trains provides a collection of must-have automated process shortcuts that we all knew we needed, but couldn't, or wouldn't, create ourselves," says Asselman.

As it happens, Theator's developers prefer to work by command-line interface rather than a web-based UI. Despite the fact the Trains offers a user-friendly UI, they were pleased to see that by leveraging Trains' set of APIs, they had the option to work completely in code, integrating it smoothly into their existing workflows. This was a benefit especially as the development team is made up of a diverse group of professionals from various backgrounds, accustomed to working in varied environments. Smoothly integrating Trains into their code meant minimal friction or changes to their preferred workflows.

Results:

Theator saw immediate savings related to reduced AWS costs as cloud-based resources were now only used for precisely the time they were needed. Allegro Trains also eliminated the need to hire a dedicated DevOps engineer to support the data science team and developers, and liberated them from the distraction that managing servers represented. Asselmann estimates the direct ML-Ops related cost savings to be around $130K-$170K annually at their current workloads. As their AI workloads grow, so will the savings.

Aside from these savings, Asselmann appreciates the significant jump in productivity due to simplified reproducibility. "For each experiment," he reports, "adding the Trains code snippet made the process 'hands-free' as developers no longer needed to document every parameter that went into the code and associated packages, then migrate these to a new machine with the data and only then run the experiment. The alternative," he explains, "means running

> Scientists who prefer code-based management, rather than via Trains' web-based UI, could easily incorporate Allegro functionality into their own code.

'fairly similar' experiments without a precise duplication process – which is not only far from best practice, but is also frowned upon in a heavily regulated industry where AI results can mean the difference between success and tragedy. These additional savings are probably on par with the ML-Ops direct savings, if not more."

> Data Scientists also benefit from complete scalability, effortless reproducibility of experiments, as well as tracking of all server metrics to track crashes and memory leaks triggered by specific processes.

Due to the deployment agnosticity of Allegro Trains, and its ability to run on-prem, on cloud or any combination thereof, Asselmann is confident that as they grow and potentially invest in their own DGX pods, Allegro Trains will grow with them seamlessly with no change to their workflows.

"If I had to sum it up," concludes Asselman, "I'd have to say that, far from dictating new processes and changing the workflows that worked for us, Allegro Trains just feels natural. Quietly and intuitively working in the background, we can focus on empowering surgeons, not wrestling with the steps to get there."