LiCO: Simplifying AI Development

Positioning Information

The compute and storage resources necessary to run AI are rapidly increasing, especially those in the deep learning space. Frequently, the growing needs of the data science team outpace the purchasing cycle – especially for large companies. This leads to challenges between the data science teams and the business teams, and the traditional approach of giving the data science team dedicated resources only exacerbates the problem.

Typically, a powerful system with multiple top-of-the-line GPUs and CPUs is deployed for each data scientist, which is effective for model training but will be overkill for most of the overall development lifecycle. These systems are also difficult to share amongst the data science team, resulting in both lower resource utilization and less efficiency for the data scientists.

To keep pace with performance demands, the business needs to continually upgrade to the best-in-class systems for their data scientists, and retire the previous systems. The result: continually paying high prices for “bleeding edge” systems which are quickly obsolete and relatively poor return on investment (ROI).

In contrast, scale-out solutions utilizing a cluster of systems can be easily expanded as both performance needs and the number of workloads grow. In this situation, the data science team is able to efficiently share a large pool of computing resources which improves utilization, efficiency, enables performance scaling, and a results in better ROI.

The only issue with a scale-out scenario is that a cluster can be hard to use for AI applications. Enter LiCO. LiCO’s intuitive interface helps simplify managing resources in the cluster for system administrators and running of AI jobs for, data scientists, and AI engineers.

LiCO Functionality for System Administrators

LiCO offers defined role types for both cluster management (administrators) and development (users). The administrator role type provides many tools to help address the management of the cluster. Upon logging into LiCO, the administrator gets a snapshot of the cluster's health. They can see the usage of the cluster (CPUs and GPUs), memory, storage, and network.

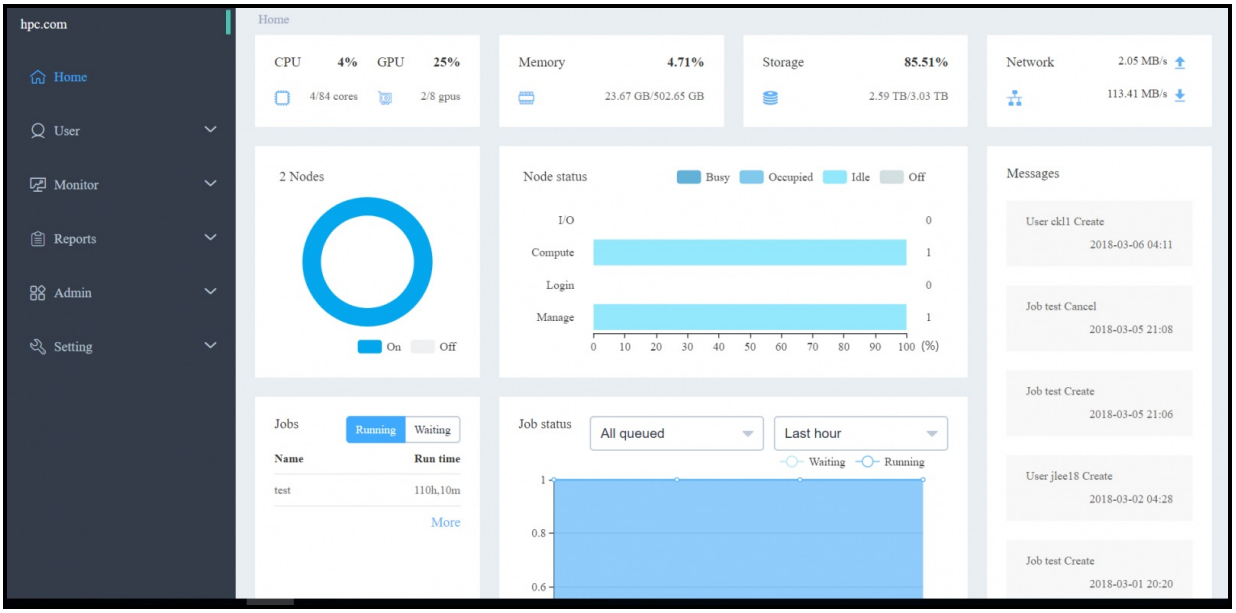

Figure 1. Administrator View for LiCO (click the image to see a larger version)

Additionally, LiCO provides a number of more detailed tools to manage users and cluster resources. The Administrator can use LiCO to manage system access by setting user groups and users. A key feature for many clients is the ability to establish billing rates for various resources and groups - for example, establishing a base cost for CPU usage time. The administrator can meter usage among various groups within the organization in an automated and quantitative fashion. For management of the physical cluster components, the administrator can drill down into a more detailed view of cluster resources, including the CPU utilization, power consumption, and GPU usage.

The AI Workflow

The AI workflow is characterized by a large number of inter-related tasks – later tasks often improve the results of earlier tasks as more data is processed. Let's explore the various stages of the AI workflow in more detail and see where and how LiCO fits in this process.

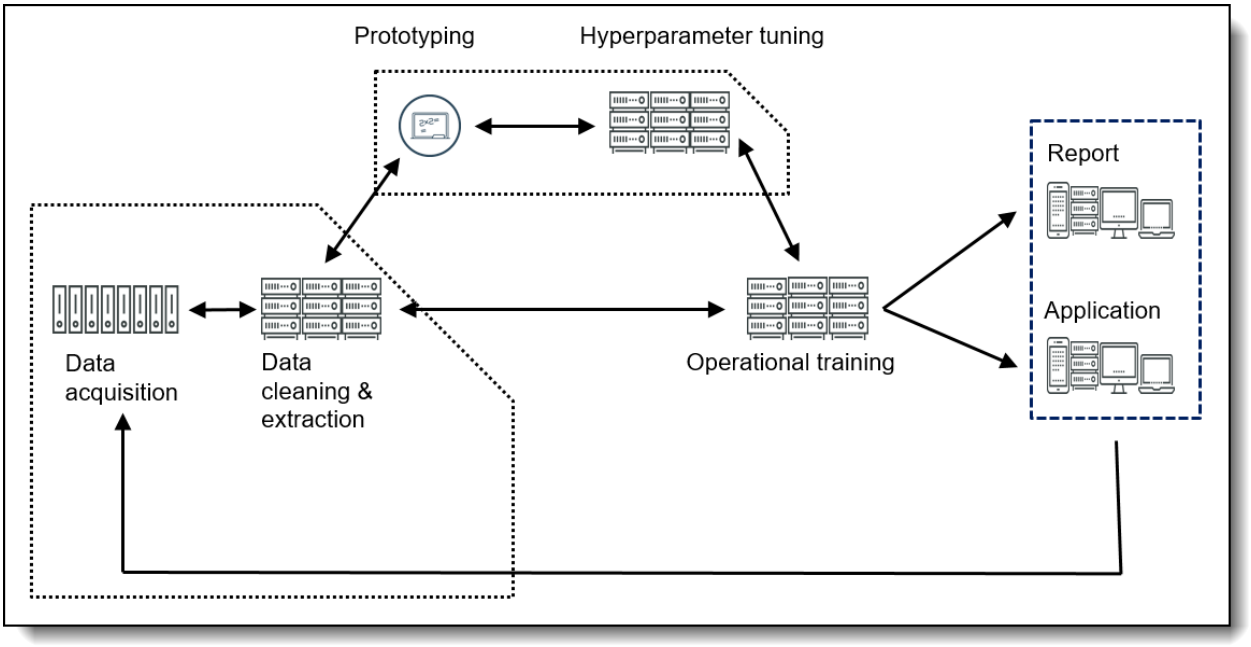

Figure 2. AI Workflow

The stages of the AI workflow are as follows:

1. Data acquisition

   The process begins with data acquisition. This is often the most challenging aspect of the workflow for companies getting started in AI. Typically, data feeds from across the organization feed a repository known as a Data Warehouse or Data Lake. For historical or trend analysis, the data collection stage requires ongoing ingestion for at least two or three years to generate reliable models.

   Additionally, this data must be collected in a manner that adheres to the rules of basic data governance to avoid incoherent, non-joinable data with unreliable quality. Typically, successful data acquisition requires either a strong cross-functional team to lead the efforts, or consistent cooperation between various C-Suite officers, so the Data Lake doesn't become a 'Data Swamp'.

2. Data cleaning and extraction

   Data cleaning and extraction are often characterized by heavy, tedious manual formatting and data adjustments. Data cleaning typically requires a domain expert, data scientists and data engineers working together to review each column of data and eliminate various potential sources of error. While it may be tempting to skip such tedious tasks, short-changing data cleaning and rushing onto the next step usually results in project failure.

   A good rule of thumb is "garbage in, garbage out" – meaning that poor quality data leads to poor quality models. Conversely, clean data can open up a wide variety of models to be developed in the next step, prototyping.

3. Prototyping

   Prototyping is where the data science team spends their time experimenting with various models. Depending on the project needs, these models can range from simple statistical models to complex deep learning models. These models tend to be rough, non-optimized solutions, requiring experiments and best guesses from the data scientists to estimate hyperparameters such as learning rate or the number of hidden layers.

4. Hyperparameter tuning

Hypertuning is where the data scientist varies different inputs to the prototype model in order to try to achieve a higher level of accuracy. This is also a time-intensive task for two reasons: there is a wide variety of possible inputs (both in terms of the number of hyperparameters and the possible range of values they may take on), and the process for finding the right set of inputs to the model is largely trial-and-error.

This process of running the model repeatedly is typically done by the data scientist, sub-optimizing the data scientist's time. This is where LiCO is especially valuable. LiCO's intuitive interface allows non-data scientists without deep technical skills to modify hyperparameters and re-run workloads.

LiCO also helps with operational training. When completing the previous tasks, data scientists and their teams tend to run the models with as many compute resources as they have access to. This is not sustainable when a model moves into operational training, where AI engineers try to balance resource usage with speed of retraining. In this step, the objective changes from "train as fast as possible" to "train using the minimal necessary resources".

For example, if a model has a Monday at 4 A.M. retraining deadline, the AI engineer will have to determine what resources are required, and when to start the job. LiCO workflow templates allow users to dedicate compute resources to a particular job, or allow the resources to be split between multiple jobs. LiCO also supports the use of queues to divide a cluster into logical groups; for example, a queue could submit jobs only to systems that contain NVIDIA V100 GPUs. This queue would be more appropriate for larger AI training jobs and less resource efficient for tasks such as image preprocessing.

5. Operational training

   The final task is using the model. This involves using an inference engine to either create reports or feed into a user application. Reports are typically generated by Business Intelligence software such as MicroStrategy or Tableau and are used to send information in a batched format to the various interested parties. User applications containing inference engines may be designed for processing streaming data or running on the edge. Most of the lifespan of successful AI projects is spent in this task – leveraging models for business value.

Managing the whole AI Lifecycle

LiCO enables the use of both CPUs and GPUs as needed during the AI workflow. Significant emphasis is placed on the parts of the AI workflow where GPUs greatly accelerate the process, such as in hyperparameter tuning and operational training. However, the remaining tasks in the workflow generally rely on CPU power. A balanced cluster that has the appropriate ratio of CPUs to GPUs will not only return the greatest ROI on the equipment but also provide a superior user experience for the AI team.

For example, when developing a new model, significant CPU power will be used in the data cleaning step, typically the second longest task in the workflow. For inferencing and reporting, the longest task of successful projects, CPUs are also almost exclusively used. LiCO uses a single interface to manage both the CPUs and GPUs in a cluster to achieve maximal usage and performance.

What LiCO is not

It can be confusing to sift through all the available data science tools to put together the set that works for your company. We are extremely proud of LiCO, (it was chosen as Best AI Product or Technology by HPCWire in 2018). However, it is not a "do-it-all" solution.

- LiCO is not intended to be a data wrangling or data management tool. While it does provide some visualization tools and metadata on the datasets used by deep learning models, this is not the primary focus of the software.
- Additionally, LiCO is not a tool for data scientists to prototype models quickly such as is commonly done in Jupyter Notebooks. Although LiCO provides workflow templates LiCO is not a data and workflow processing tool such as Apache Spark.

- LiCO does not support streaming data and therefore is not a substitute for tools such as Apache Kafka.
- Finally, although LiCO provides some inferencing support, it is not an AI deployment tool.

Although LiCO does not handle these pieces of the AI workflow, it is designed to work with all of the technologies to be an essential part of a complete solution. Overall LiCO is a tool that simplifies hyperparameter tuning and operationalization of deep neural networks to help you turn move prototypes to production as effectively as possible.

Popular AI Frameworks

The following frameworks are among the most commonly used tools for programming deep learning applications:

- TensorFlow is currently the most popular deep learning framework and is available via both a Python and R interface. TensorFlow can support both CPUs and GPUs in either multi-GPU or multi-node formats.
- Caffe is a well-known framework which was developed by the Berkeley Vision and Learning Center (BVLC). It focuses on image classification problems and operationalized deep learning workloads. Caffe supports CPUs and multiple GPUs within a single node.
- MXNet was developed as a collaboration between many universities and companies including Amazon and Microsoft. MXNet supports numerous types of deep learning models and is designed to be distributed on a dynamic cloud infrastructure (though it can also run in on-premises). It can run in CPUs and GPUs in either multi-GPU or multi-node formats.

LiCO supports all of these frameworks. The next section discusses how the templates available for these frameworks simplify the deployment of AI workloads.

LiCO Functionality for Data Scientists and AI Engineers

Data scientists and AI engineers using LiCO to run training and inference workloads will have the user role type and have access to the User home screen. The User home screen provides an overview of resource usage, showing the status and number of jobs, CPU and GPU utilization, memory usage, and network speed.

Containerized environment

Managing the environment needed to run both machine learning and deep learning applications can be a major challenge. Especially when the user attempts to do this in a multi-tenant environment, simply getting a job to run in the correctly configured environment proves problematic. To solve this problem LiCO leverages Singularity. The user can download any of a number of popular Singularity containers from Singularity Hub or using a single pull command can import Docker containers into Singularity from Docker Hub. This is a powerful tool that allows data scientists and AI engineers to update AI frameworks, add new frameworks quickly, and effectively manage a multi-tenant environment.

Job templates

Users also have the ability to create and run job templates. The TensorFlow Multinode template is shown below:

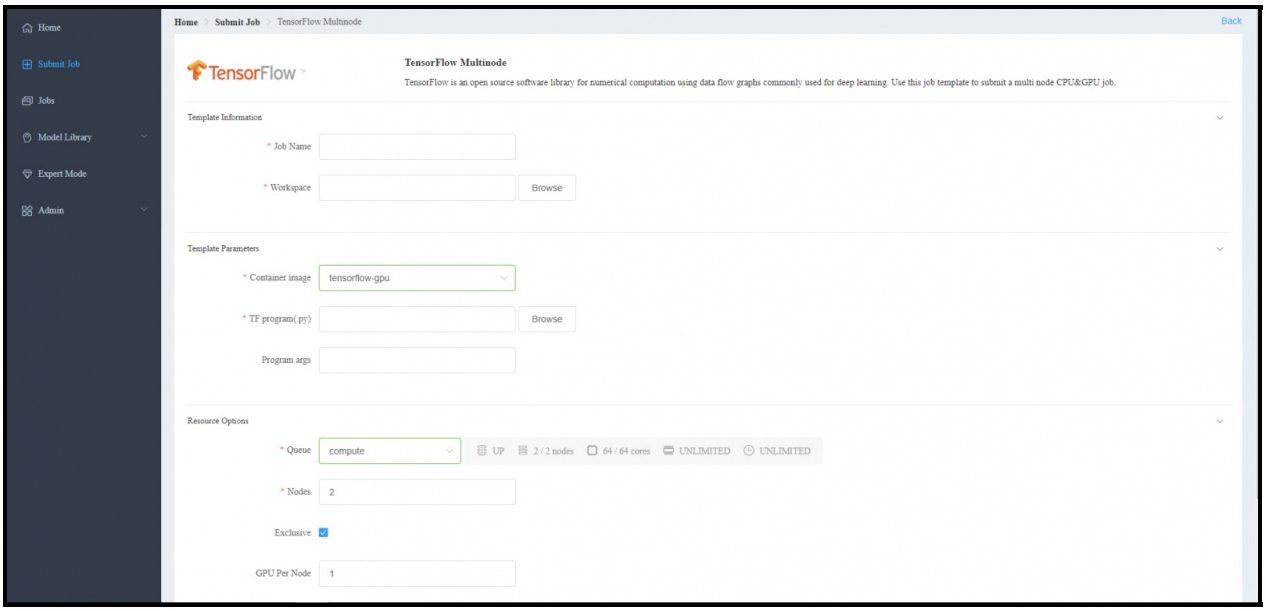

Figure 3. Template to run TensorFlow (click the image to see a larger version)

Job templates allow users to run AI & HPC workloads in a simplified manner. For example, the TensorFlow Single Node template, which enables the running of AI jobs written with TensorFlow to run on a single node. This template is useful for fast prototyping of new deep learning models - simply select the code, the containerized environment, CPUs or GPUs, and the template will run the job. Another standard template is TensorFlow Multinode, which allows for the distributed training of TensorFlow jobs. This template is useful for more developed models that have been programmed to run in a multi-node format. After running a TensorFlow job (using either Single Node or Multinode template), the user has ability to view the logs and TensorBoard (if created within their code) from within LiCO to view commonly recorded metrics such as accuracy and loss.

TensorFlow is just one of the many job templates available within LiCO. Other AI job templates include Caffe, Intel Caffe, MXNet Single Node, MXNet Multinode, and Neon. These job templates allow users access to most of the popular AI frameworks within LiCO, without having to manually configure the software on the systems. For HPC users, job templates include popular workloads such as MPI, ANSYS, and COMSOL. Additionally, there are options to submit shell scripts and SLURM jobs or even make custom templates with custom parameters to satisfy the needs of any use case.

LiCO provides additional functionality for Caffe users in the form of a workflow template. This covers additional aspects of the testing and development lifecycle not covered in the job templates discussed above. The user can begin by uploading a dataset in which they can view the division of the data into training, testing, and validation datasets and the class balance. Next, the user can view and edit network topologies written in Caffe, such as AlexNet and LeNet. Additionally, in order to confirm the edited network topology, this template offers a tool that allows the user to conveniently visualize the network. Finally, there is a models section, in which the user can view previously run models and see if they were successful or not.

Monitoring capabilities

If the user clicks into the Caffe model, they additionally can view the training accuracy, loss, and processing speed by epoch number (see below). This allows the user to see the effectiveness of his model in real time. From this screen, the user can also quickly re-run the model or alter the hyperparameters as needed. This allows the data scientist to write the base model, and then pass off the hyperparameter tuning to a junior data scientist or analyst. This workflow template provides the framework with which to modify many common hyperparameters such as learning rate, decay rate, and regularization type. LiCO also automatically records the experiments via job tracking, saving valuable time.

For more developed models that are able to run in multi-node format, the AI engineer can re-run jobs in order to determine the appropriate resources to allocate for the retraining of AI workloads.

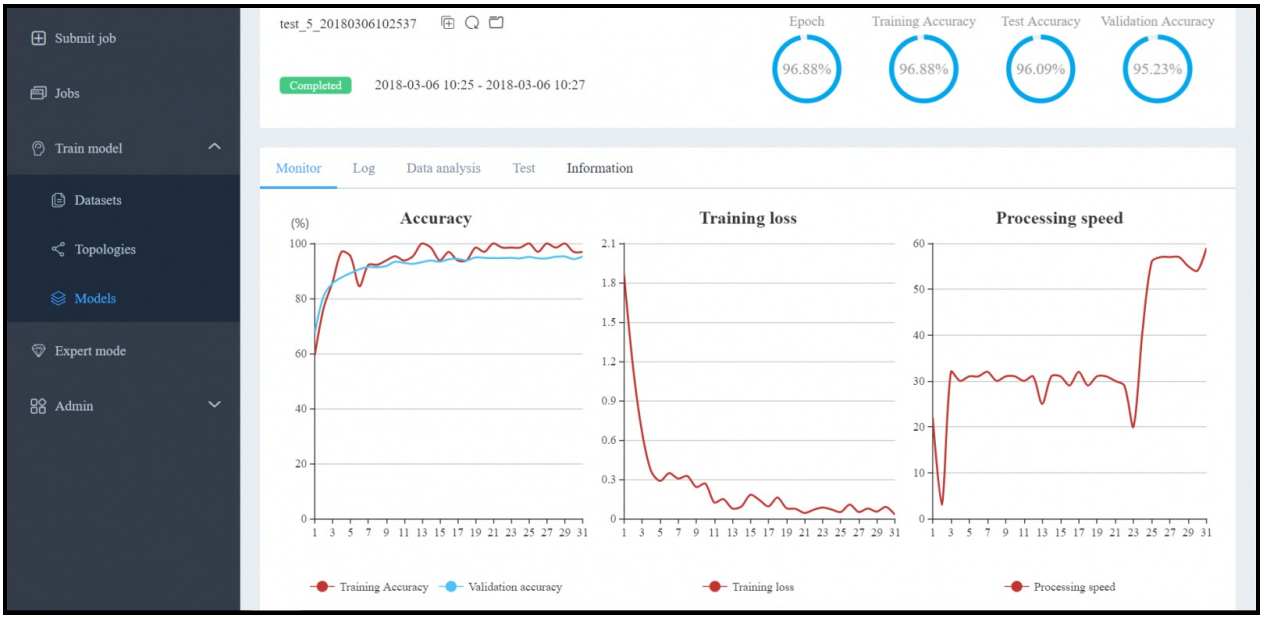

Figure 4. Caffe training accuracy, loss, and processing speed tracked in real time with LiCO (click the image to see a larger version)

No-code templates

Perhaps its most helpful feature, LiCO also provides a number of no-code templates called Lenovo Accelerated AI. These Lenovo Accelerated AI templates allow the user to run common AI use cases for image recognition and Natural Language Processing (NLP). The common image-based AI use cases supported by Lenovo Accelerated AI are:

Image Classification is a task in which a dataset containing images of primarily one class (e.g. a plane, a dog, a car) is provided to the model, and the AI model determines the class of that image. Popular datasets such as ImageNet are processed primarily via image classification workloads.

Object Detection is a step beyond image classification, in that it not only identifies the class (e.g. a plane, a dog, a car) but also identifies a region of interest or bounding box around that class. This allows object detection algorithms to identify multiple classes within the same image.

Image Segmentation goes a step further and divides the entire image, on a pixel by pixel basis, into classes. In this case, not only the main objects are detected, but also the background objects (e.g. grass, sand, road). This becomes increasingly important as algorithms are attempting to understand the context and situation within an image.

The next template, Image GAN, is a Deep Convolutional Generative Adversarial Network that is used to create images of the desired class. In the Image GAN, there are two networks – one that generates images, and one that judges those images. After many iterations, the generated images begin to create pictures of the desired class.

The final image-based template, Medical Image Segmentation, performs application-specific image segmentation for the health care and research fields.

There are also two non-image-based templates included with Lenovo Accelerated AI – Seq2Seq and Memory Network. Seq2Seq is commonly used to translate from one language to another. This is done through the use of a recurrent neural network composed of a specific type of nodes called Long Short-Term Memory nodes. The other template, a Memory Network, frames the NLP as a question and answer the problem and is useful for applications such as chatbots.

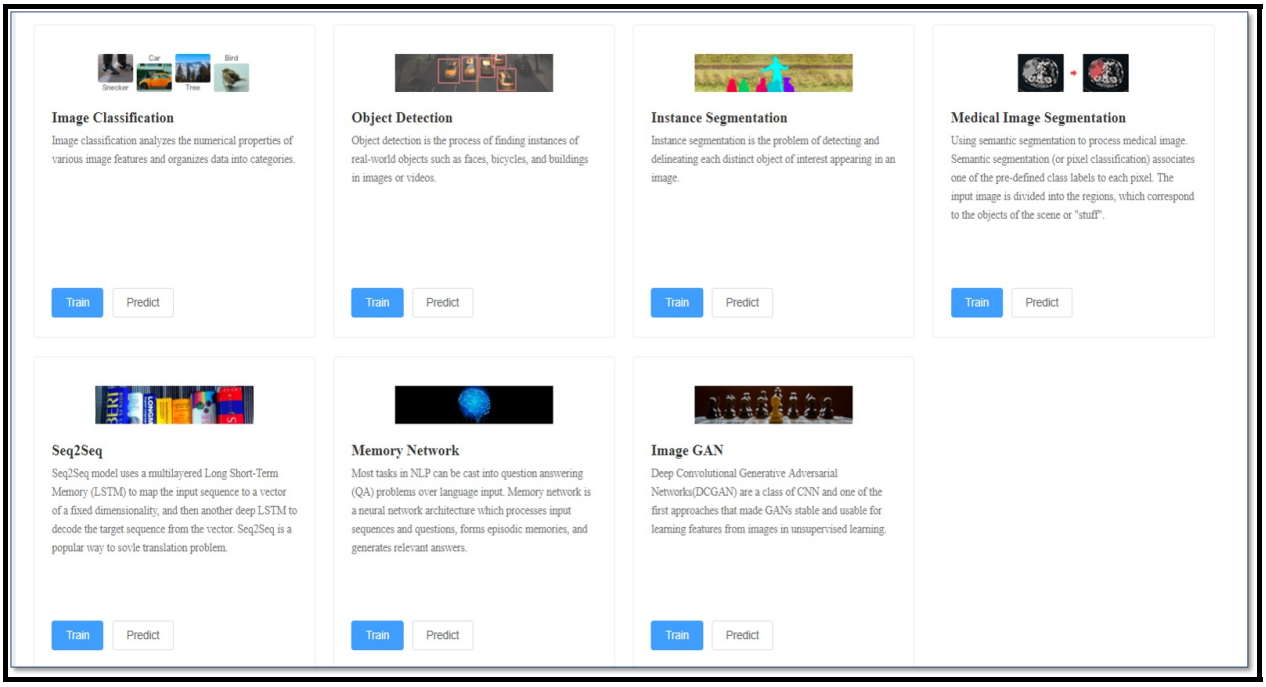

Figure 5 Lenovo Accelerated AI templates (click the image to see a larger version)

With Lenovo Accelerated AI templates, LiCO greatly reduces the barriers to entry to perform AI experimentation. Non-technical users or those just beginning their AI journey can perform training or inference on datasets, without the need to code. This further increases the ability to utilize the HPC cluster and gain value from the hardware ecosystem.

Conclusion

Lenovo Intelligent Computing Orchestration (LiCO) provides wide-ranging functionality to enable the deployment of AI workloads on HPC systems.

For system administrators, LiCO provides sophisticated hardware monitoring and management, as well as tools such as billing groups to manage usage within organizational structures. LiCO's queue management functionality also allows administrators to divide compute resources for different workloads.

For data scientists and AI engineers, LiCO's job and workflow templates simplify the deployment of AI workloads, allowing for fast hyperparameter tuning and workload optimization.

Finally, Lenovo Accelerated AI supports users with a limited technical background in AI by providing easy-to-use templates that can perform training or inference without the need to code.

Additional information

The following resources provide additional information about LiCO:

- LiCO Website
  https://www.lenovo.com/us/en/data-center/software/lico/
- LiCO Product Guide
  https://lenovopress.com/lp0858-lenovo-intelligent-computing-orchestration-lico
- LiCO Datasheet
  https://lenovopress.com/datasheet/ds0029-lenovo-intelligent-computing-orchestrator
- LiCO User Guide
  https://download.lenovo.com/servers_pdf/LiCO_5.2.1_User_Guide_20190104.pdf
- LiCO Support Page
  https://datacentersupport.lenovo.com/us/en/products/solutions-and-software/lenovo-intelligent-computing-orchestration/lico/solutions/HT507011
- Lenovo Artificial Intelligence
  https://www.lenovo.com/us/en/data-center/solutions/analytics-ai/

About the author

David Ellison is the Senior Artificial Intelligence Data Scientist for Lenovo. Through Lenovo's US and European Innovation Centers, he uses cutting-edge AI techniques to deliver solutions for external customers while internally supporting the overall AI strategy for the World Wide Infrastructure Solutions Group. Currently, his emphasis is in distributed training of neural networks and fine-grain objection detection using high-resolution imaging. Previous to Lenovo, he ran an international scientific analysis and equipment company and worked as a Data Scientist for the US Postal Service. David has a PhD in Biomedical Engineering from Johns Hopkins University.

Related product families

Product families related to this document are the following:

- Artificial Intelligence
- High Performance Computing

Notices

Lenovo may not offer the products, services, or features discussed in this document in all countries. Consult your local Lenovo representative for information on the products and services currently available in your area. Any reference to a Lenovo product, program, or service is not intended to state or imply that only that Lenovo product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any Lenovo intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any other product, program, or service. Lenovo may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

Lenovo (United States), Inc.
8001 Development Drive
Morrisville, NC 27560
U.S.A.
Attention: Lenovo Director of Licensing

LENOVO PROVIDES THIS PUBLICATION ”AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. Lenovo may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

The products described in this document are not intended for use in implantation or other life support applications where malfunction may result in injury or death to persons. The information contained in this document does not affect or change Lenovo product specifications or warranties. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of Lenovo or third parties. All information contained in this document was obtained in specific environments and is presented as an illustration. The result obtained in other operating environments may vary. Lenovo may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any references in this publication to non-Lenovo Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this Lenovo product, and use of those Web sites is at your own risk. Any performance data contained herein was determined in a controlled environment. Therefore, the result obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

This document, LP1081, was created or updated on February 27, 2019.

Send us your comments in one of the following ways:

- Use the online Contact us review form found at:
  https://lenovopress.lenovo.com/LP1081
- Send your comments in an e-mail to:
  comments@lenovopress.com

This document is available online at https://lenovopress.lenovo.com/LP1081.

Trademarks

Lenovo and the Lenovo logo are trademarks or registered trademarks of Lenovo in the United States, other countries, or both. A current list of Lenovo trademarks is available on the Web at https://www.lenovo.com/us/en/legal/copytrade/.

The following terms are trademarks of Lenovo in the United States, other countries, or both:
Lenovo®

The following terms are trademarks of other companies:

Intel® is a trademark of Intel Corporation or its subsidiaries.

Microsoft® is a trademark of Microsoft Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.