DeepForge: A Machine Learning Gateway for Scientific Workflow Design

Brian Broll (Co-PI)

Akos Ledeczi (PI) · Umesh Timalsina · Peter Volgyesi · Tamas Budavari

https://deepforge.org

PROJECT GOALS

The goal of this research is to develop a Software as a Service platform for applying deep learning within diverse scientific domains that integrates with existing cyberinfrastructure.

This platform is also designed to promote **simplicity**, **collaboration**, and **reproducibility**. These design goals are supported through:

- Data provenance (including trained model artifacts)
- Integration with existing cyberinfrastructure
  - Computational resources such as Slurm and PBS
  - Data resources such as Globus Connect
- Real-time collaborative editing and integrated version control
- Extensible APIs

BACKGROUND

TensorFlow

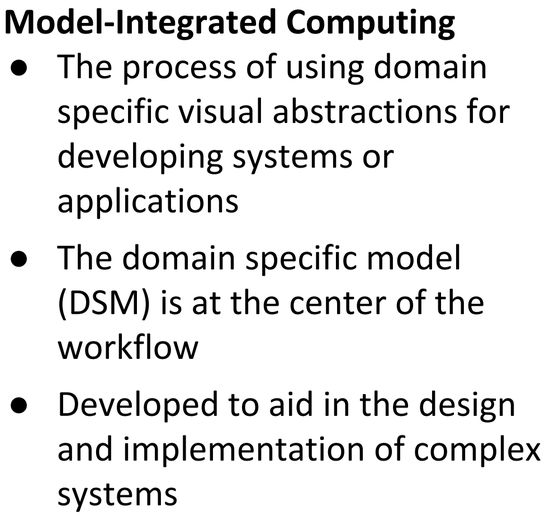
- Open source machine learning framework in Python
- Supports many different deployment platforms from clusters to mobile devices

Model-Integrated Computing

- The process of using domain specific visual abstractions for developing systems or applications
- The domain specific model (DSM) is at the center of the workflow
- Developed to aid in the design and implementation of complex systems

WebGME

- An MIC framework for creating domain specific development environments
- Cloud-based infrastructure
- Provides a number of useful features including version control and collaborative editing



CORE CONCEPTS

- Core Concepts are the components of the DSML
- **Operation** - an atomic function with variable number of inputs and outputs as well as external parameters and references to other artifacts.
- **Job** - a running **Operation** with execution information
- **Pipeline** - Directed Acyclic Graph (DAG) of **Operations** representing an experiment
- **Execution** - A running **Pipeline** with execution information
- Supports extension for additional domains

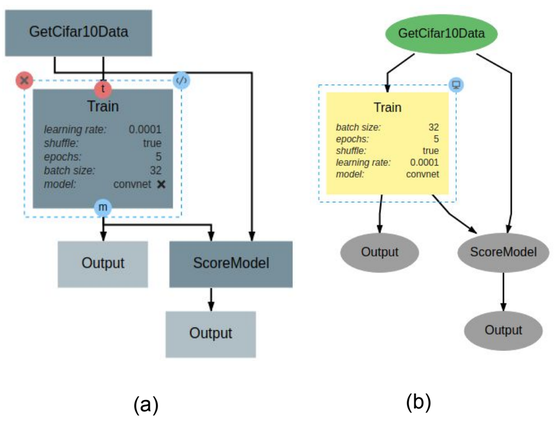


Figure 3: (a) Editing a training and testing pipeline on CIFAR 10. (b) Monitoring the execution of the pipeline in (a)

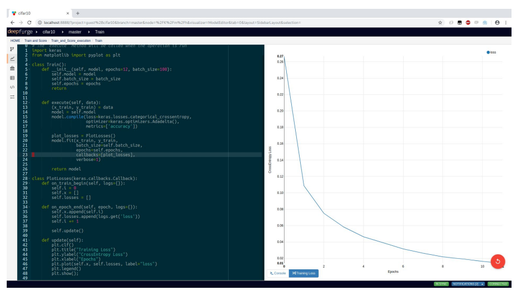


Figure 2: Viewing a training job with real-time plotting feedback

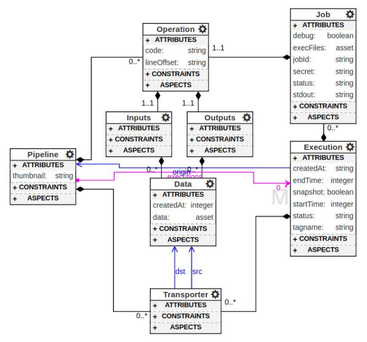


Figure 4: Formal specification of the language used in the DSM based on the provided core concepts

PLATFORM

General

- Developed using WebGME
- Domain Specific Modeling Language (DSML) formalized from the Core Concepts (Figure 4)
- Supports extension with other deep learning libraries (or domain models)

Reproducibility

- Automatic version control for data, code and parameters guarantees that every state is reproducible
- Automatically tag branches when executing training or testing pipeline
- Data provenance for data and trained models

Extensibility and Infrastructure Integration

- Supports computational resource integration via compute adapters
- Storage resource integration is supported through storage adapters
- Existing integrations include SciServer, S3, and individual compute workers

Productivity and Accessibility

- Visual editors for architectures and pipelines; textual editors for operation implementations
- Design-time error detection and dimensionality information for neural network architectures
- Execution on connected computational resources
- Collaborative editing support
- Executing a training/testing pipeline caches intermediate data allowing individual jobs to be re-run without recomputation

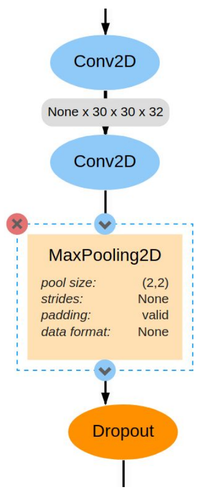


Figure 5: Editing a neural network architecture using the visual editor.

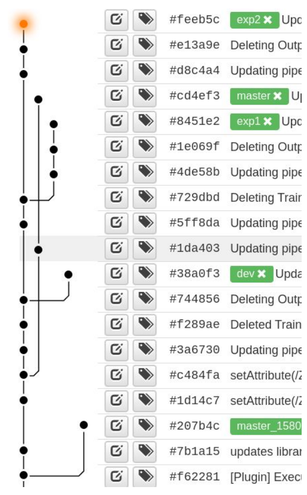


Figure 6: Integrated version control makes any historical state reproducible and facilitates collaboration

SUMMARY

DeepForge is an open source platform for deep learning designed for promoting reproducibility, simplicity and rapid development within diverse scientific domains. This includes:

- Leveraging strengths of Model-Integrated Computing to design a gateway for deep learning
- Collaborative editing capabilities
- Integrated version control of code and data
- Distributing jobs over connected computational resources and integration with existing infrastructure

deepforge

ONGOING RESEARCH

Current research consists of extending infrastructure integration capabilities, improving extensibility and scriptability via APIs, and community development. This includes:

- Additional compute infrastructure integration
  - Slurm
  - CWL
- Additional storage infrastructure integration such as Globus
- Development of rich Python and JavaScript APIs
  - Pipeline creation and execution
  - Execution monitoring
  - Artifact creation and retrieval
- Maintaining a public deployment and developing community through hackathons and collaboration

This material is based in part upon work supported by the National Science Foundation under Grant Number SI2-SSE #1740151