Institute for Software Integrated Systems
Vanderbilt University

DeepForge:
A Scientific Gateway for Deep Learning

Brian Broll
brian.broll@vanderbilt.edu

Miklos Maroti, Peter Volgyesi, Akos Ledeczi

Overview

- Background
  - Deep Learning
  - Model Integrated Computing
- Core Concepts
- Platform
  - Design Goals
  - Architecture
- Future Work
- Demo

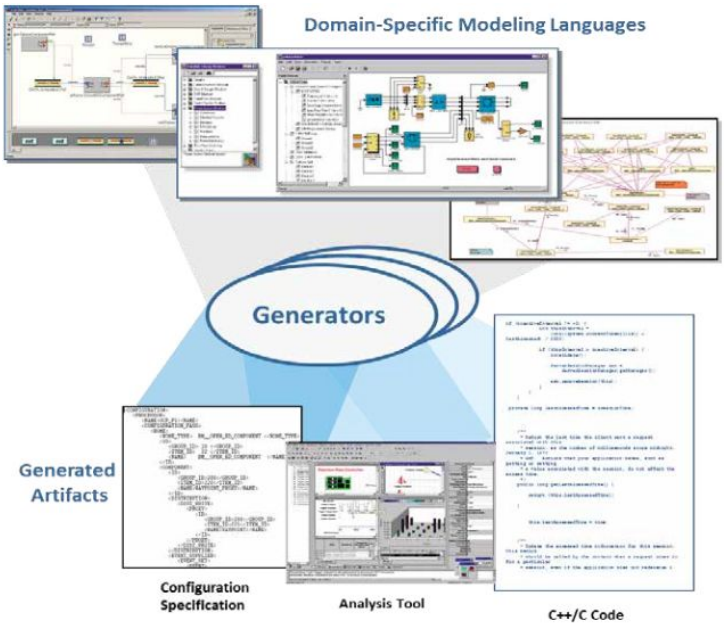
Background

Deep Learning

- A *deep neural network* is an artificial neural network with multiple hidden layers
- Flexible enough to be applied to a number of problems:
  - Speech-to-text
  - Image segmentation
  - Image classification
  - Learning embeddings
  - Styling images
  - etc
- State of art in almost everything

Model Integrated Computing

- The process of using domain specific abstractions for developing systems or applications
- The domain specific model is at the center of the workflow
- Aids in the design and implementation of complex systems



Keras + Tensorflow

- Keras
  - High-level neural networks API in Python
  - Frontend for multiple deep learning frameworks

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

WebGME

- An MIC framework for creating domain specific development environments
- Meta-configurable (provides a modeling language for creating modeling languages - similar to UML)
- Provides a number of useful features including version control and collaborative editing
- More information available at https://webgme.org

Core Concepts

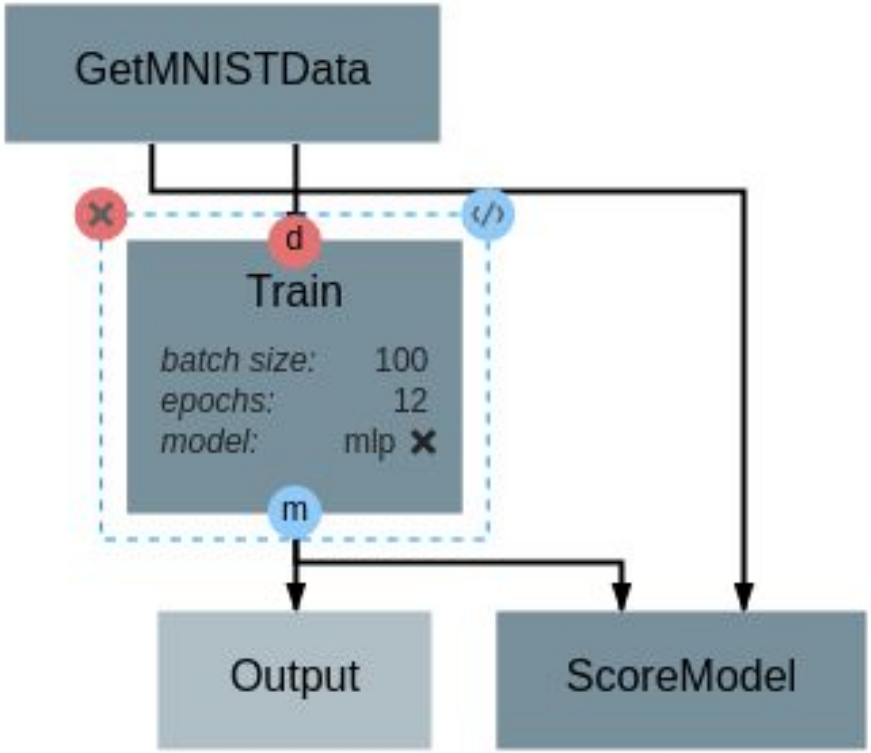
Core Concepts

- Two different types of concepts:
  - Concepts for creating executable pipelines
  - Concepts for designing neural network architectures
- Four main concepts are creating executable pipelines
  - Operations
  - Pipelines
  - Jobs
  - Executions
- Two high-level concepts for designing neural networks architectures
  - Architectures
  - Layers

Pipeline Concepts

- **Operations** are functions with multiple, named inputs and outputs
- Attributes and references can be set at design time to specify configurable parameters
  - For example, *iterations* may be specified in a training operation
- **Pipelines** represent some machine learning task composed of operations
  - Examples include training, prediction, or data augmentation
- Pipelines can also contain *Input* and *Output* operations to specify inputs/outputs of the entire pipeline

Example Pipeline



Pipeline Concepts

- **Jobs** are executable operations which contain the operation definition and metadata about the execution
  - This includes console output and plots
- **Executions** represent an executable instance of a pipeline (composed of jobs)

Platform

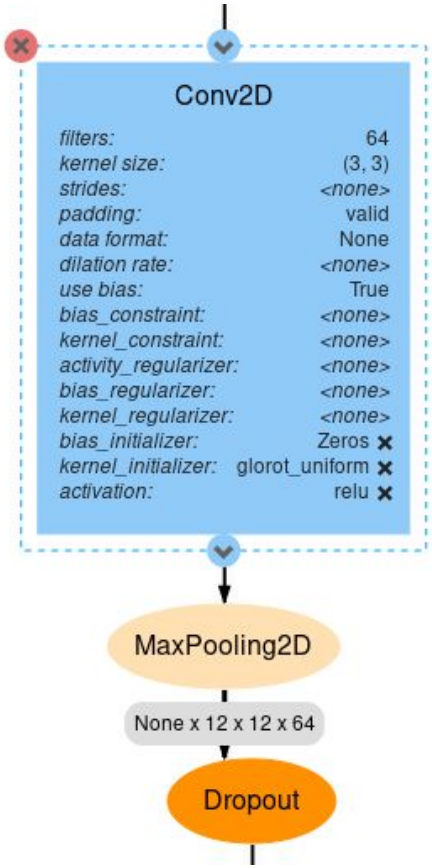
Platform

- Web-based platform with 3 driving design goals:
  - Accessibility
  - Rapid Development
  - Reproducibility
- Developed using WebGME

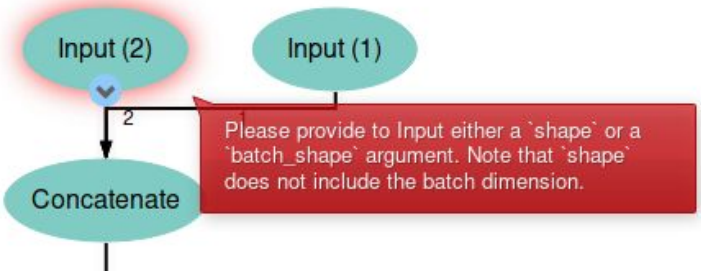
Accessibility

- Goal: Make deep learning easily accessible to other domains
- Enforce domain semantics (and constraints)
- Immediate neural network validation and feedback
  - Error feedback
  - Dimensionality information
- Utilize a hybrid visual-textual interface
  - Use the appropriate modality for the given task

Enforcing Domain Semantics: Building a Neural Network

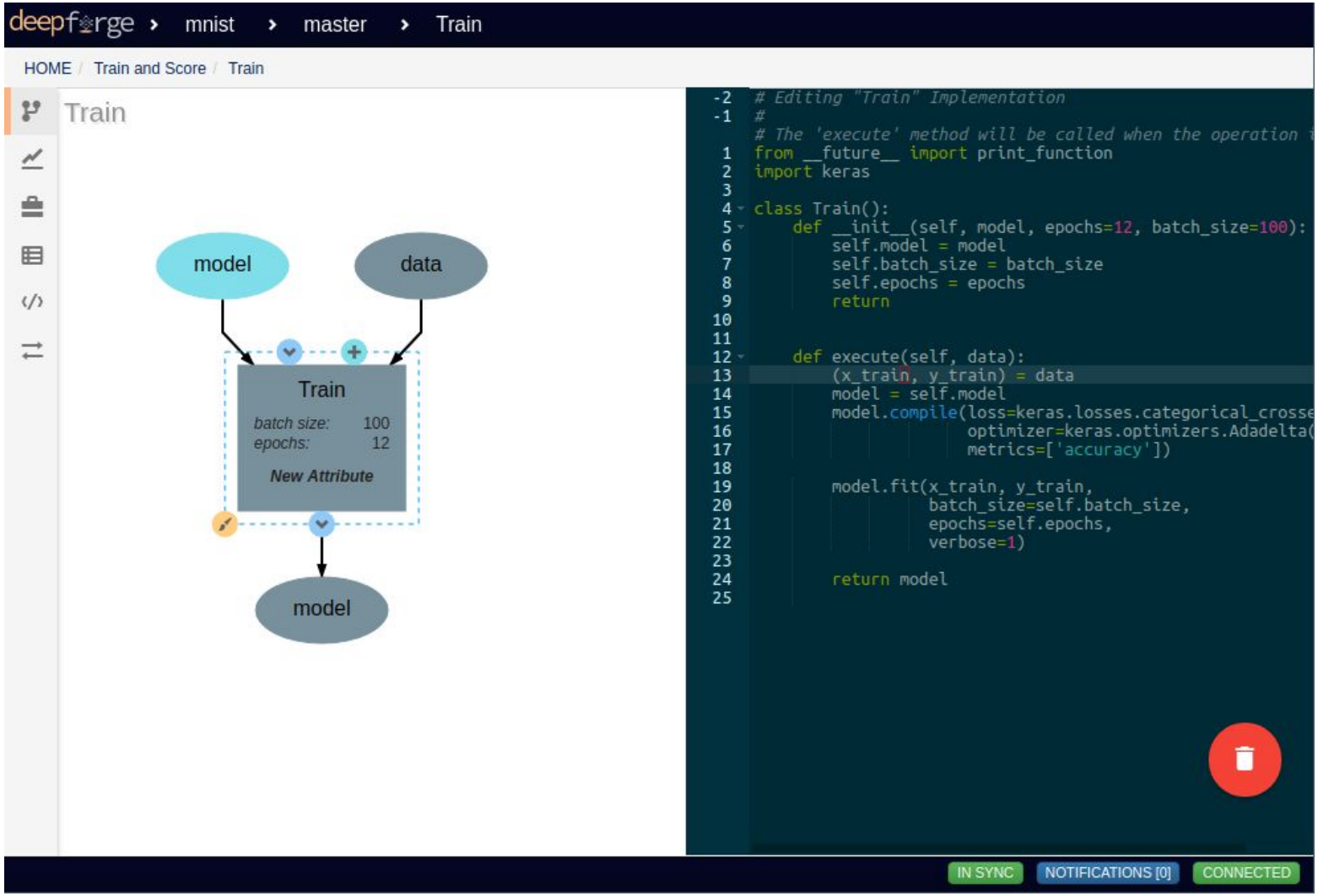


Dimensionality Feedback



Error Messages (During Construction)

Creating a Custom Operation

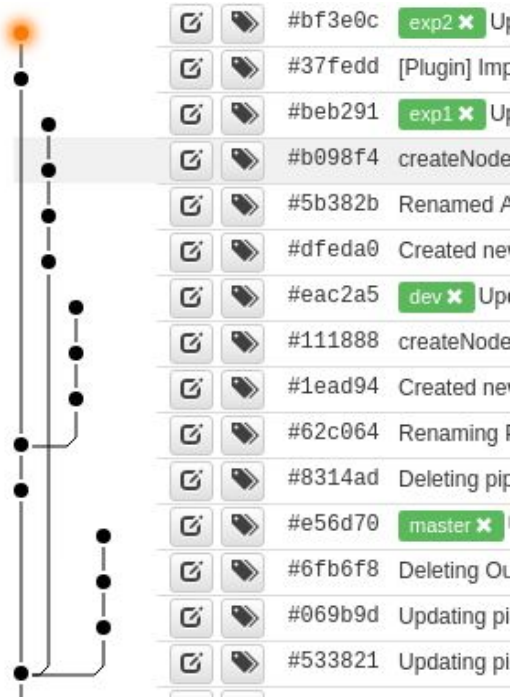


Rapid Development

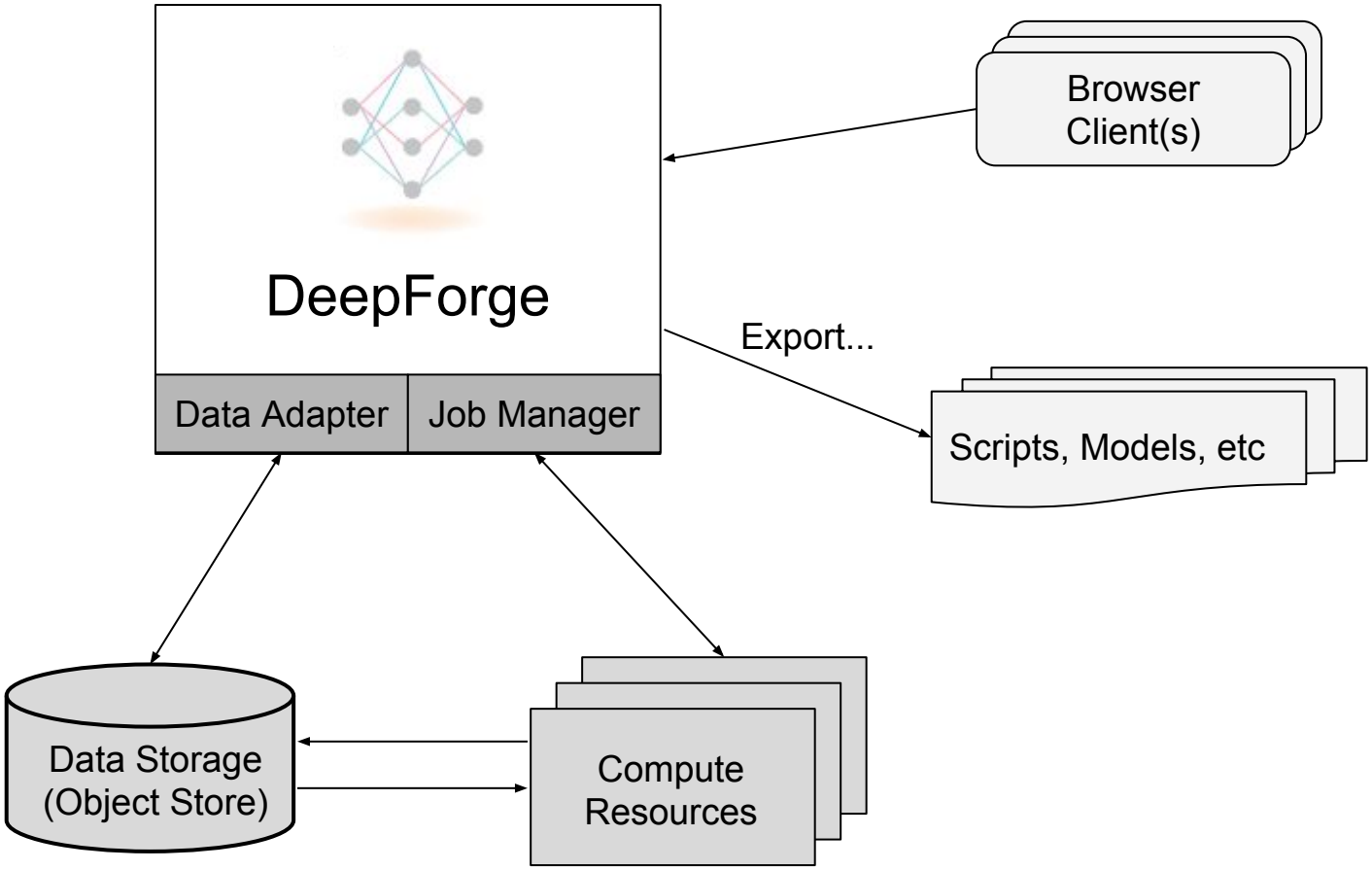
- Productivity during development was another key design decision
- Collaboration capabilities
  - Real-time collaborative editing
  - Integrated version control allows for working on isolated branches
- Support the entire development lifecycle
  - From initial creation to execution and subsequent iterations
  - Execute and monitor pipelines from the browser

Reproducibility

- Integrated version control of project content
- Blob data is referenced by hash in the project content
  - This includes datasets, trained models, etc
- Both data and the code are reproducible
- Automatic tagging commits when executing pipelines



Architecture Overview



Future Work

- Add integrations with scientific infrastructure
  - data sources
  - computational resources
- Registry for hosting project resources
  - operation definitions
  - architectures
  - trained models
- Add more data introspection utilities
- Model introspection utilities
- Architecture editor improvements
  - Parameterization
  - Composable

Demo

Questions?

- Related Resources and Links:
  - Website: http://deepforge.org
  - Source Code: https://github.com/deepforge-dev/deepforge
  - Slack Channel: https://slack.deepforge.org