# deepf@rge A MACHINE LEARNING GATEWAY FOR SCIENTIFIC WORKFLOW DESIGN

Akos Ledeczi (PI) · Brian Broll · Peter Volgyesi · Tamas Budavari https://deepforge.org

# **PROJECT GOALS**

Provide **Software as a Service** platform for applying deep learning within diverse scientific domains that integrates with existing cyberinfrastructure with emphasis on:

- **Simplicity,** web-based, visual editing, libraries
- Collaboration, real-time, versioning
- **Reproducibility,** data provenance

# BACKGROUND

**TensorFlow – o**pen source machine learning framework **Model-Integrated Computing –** domain specific languages **WebGME** – online collaborative modeling tool



	C		#bf3e0c	exp2 🗙 🛛 U
•	ß	۲	#37fedd	[Plugin] Imp
•	ß	۲	#beb291	exp1 🗙 U
	Ø	۲	#b098f4	createNode
	ß	۲	#5b382b	Renamed A
	ß	۲	#dfeda0	Created ne
	Ø	۲	#eac2a5	dev 🗙 Up
	ß	۲	#111888	createNode
	ß	۲	#1ead94	Created ne
	ß	۲	#62c064	Renaming
	C	۲	#8314ad	Deleting pip
	ß	۲	#e56d70	master 🗙
	Ø	۲	#6fb6f8	Deleting Ou
	C	۲	#069b9d	Updating p
	ß	۲	#533821	Updating p

Integrated version control makes any historical state reproducible and facilitates collaboration

Editing a neural network architecture using the visual editor



# **CORE CONCEPTS**

**Operation** - an atomic function, accepts inputs and produces outputs

- Accepts named inputs and produces named outputs
- Attributes are operation parameters set at design time
- *References* are pointers to another artifact in the DSM

### Job - A running Operation

Contains information about the execution of an operation

Includes metadata such as plots and images

# **Pipeline** - Directed Acyclic Graph (DAG) of **Operations** Represent a machine learning experiment

May include "Input" and "Output" operations

### **Execution** - A running **Pipeline**

Created when running a pipeline

Maps every operation in the origin pipeline to a job Additional concepts provided for working with neural networks: Architectures and Layers



Editing a training operation with *net* and *criterion* references and a data input, trainset



Editing a training and testing pipeline on CIFAR 10.





Extensible computational resource integration • Extending the existing architecture to support modular computational resource integration

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

## **Computational Cybersecurity in Compromised Environments**

2018 Fall Workshop | September 17-19 | Atlanta, Georgia



# **ONGOING RESEARCH**

		•••• • • • • • •	Jame .
		4- C fi http://digitaliwasoning.com/toephorps	) <b>#</b> .
prop ×		Jenny ClFAR_10 > master	
http://digitalreasoning.com/deepforge		= p	
e > CIFAR_10 > master			
n_execution_36 train			
ing "train" Implementation	Testing acquacy	nno accuracy	
ned variables: (der (Provider) prion Following will be executed when the operation is run: rning something like:	95.89 85 80 75	DownloadClifA	
<pre>t = -Model&gt; deepforge.Graph('CIFAR-10 Accuracy') gline = graph:line('Training accuracy') line = graph:line('Testing accuracy') ttributes t = 'cuda'</pre>	70 65 60 MWW	test Oop	•
<pre>image 'cudnn' 'image' 'optim' optim' ata augmentation module BatchFlip,parent = torch.class('nn.BatchFlip', 'nn.Module'] ion BatchFlip:init() ent. init(self)</pre>	55 50 43 40	statest Descutors Generato Des Mation, 7 200206 K Addres,7 400016 K Mation,7 600014 K	Execute Positive ()
<pre>f.train = true ion BatchFlip:updateOutput(input) self.train then ocal bs = input:size(1) ocal flip_mask = torch.randperm(bs):le(bs/2) or i=1,input:size(1) do if flip_mask[i] == 1 then image.hflip(input[i], input[i]) (</pre>	35 36 37 37 37 30 30 30 30 30 30 30 30 30 30		
Segmentine (Segmentine Engenerative			
Segarovite Segarovite Segarovite	3		0
			NUME SOMOTONIC COMMUNE (***

- Improved neural network architecture support
- Sequence-to-sequence prediction
- Auxiliary classifier generative adversarial networks
- Bidirectional recurrent neural networks
- Architecture analysis and feedback
- Error detection
- Dimensionality information

