torch

A dynamic view of the deep learning world

Soumith Chintala
Facebook AI Research

Overview

- What is Torch?
- The Community
- Today’s AI
- What Next?

What is torch ?

- Interactive Scientific computing framework

Strings, numbers, tables - a tiny introduction

```
In [ ]: a = 'hello'

In [ ]: print(a)

In [ ]: b = {}

In [ ]: b[1] = a

In [ ]: print(b)

In [ ]: b[2] = 30

In [ ]: for i=1,#b do -- the # operator is the length operator in Lua
            print(b[i])
        end
```

What is torch ?

- Interactive Scientific computing framework

Tensors

In [ ]:
```
a = torch.Tensor(5,3) -- construct a 5x3 matrix, uninitialized
```

In [ ]:
```
a = torch.rand(5,3)
print(a)
```

In [ ]:
```
b=torch.rand(3,4)
```

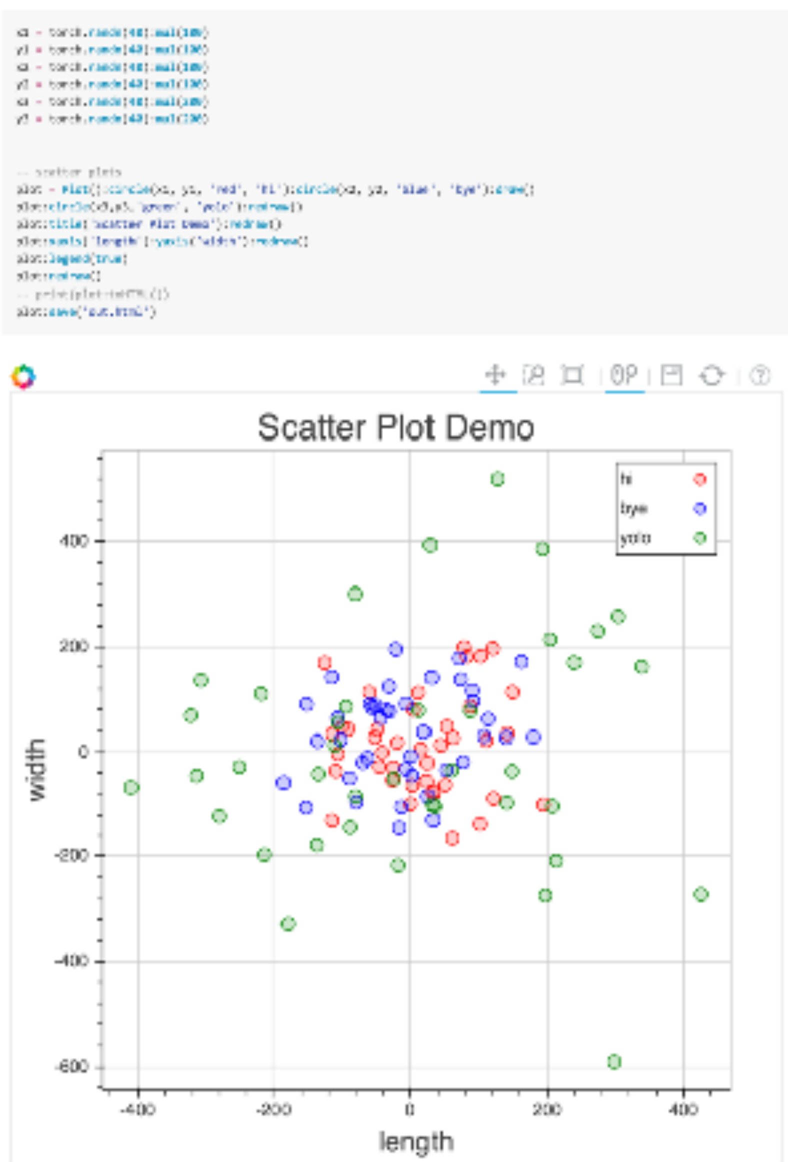
In [ ]:
```
-- matrix-matrix multiplication: syntax 1
a*b
```

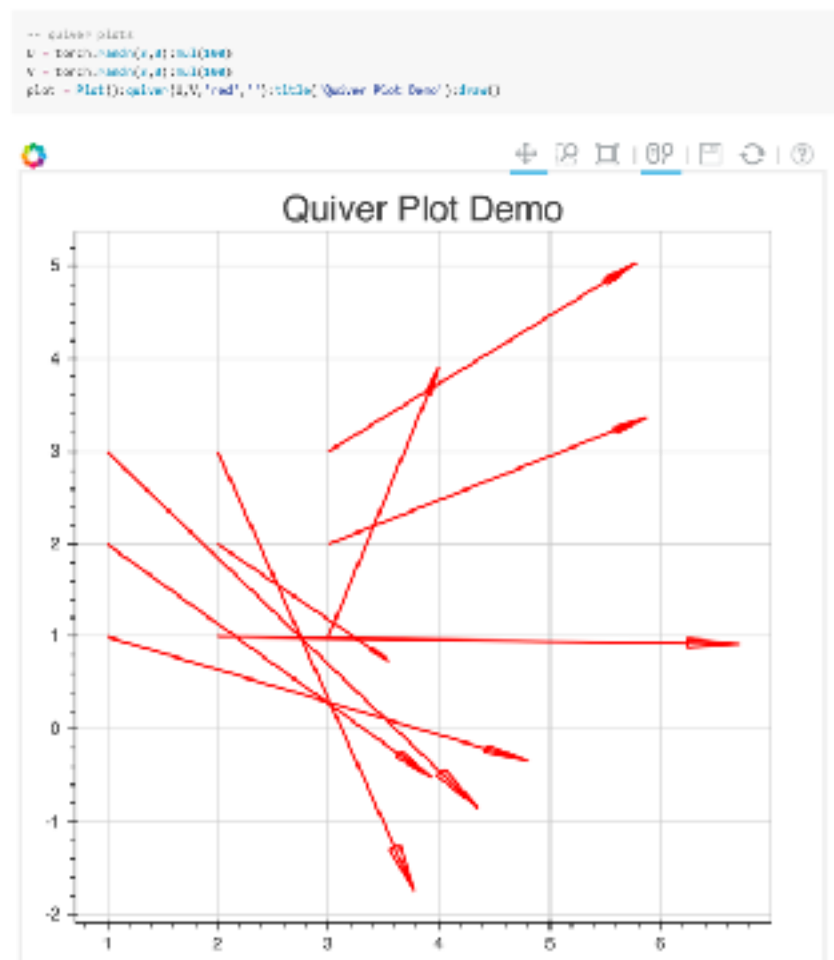
In [ ]:
```
-- matrix-matrix multiplication: syntax 2
torch.mm(a,b)
```

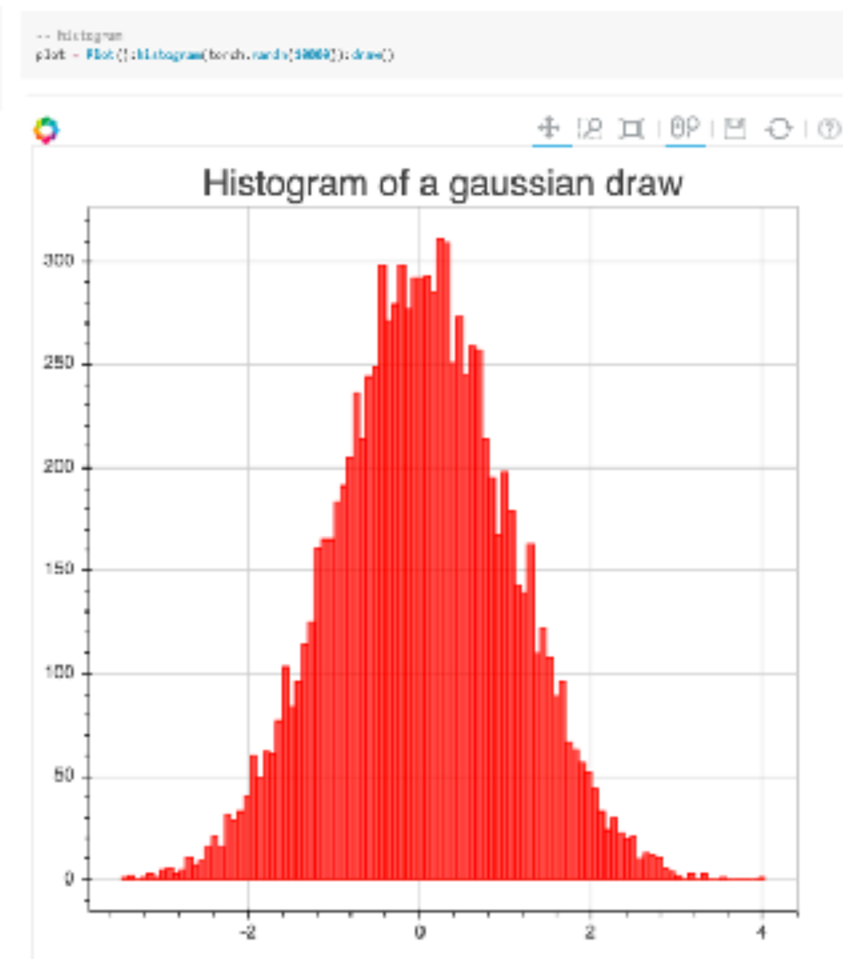
In [ ]:
```
-- matrix-matrix multiplication: syntax 3
c=torch.Tensor(5,4)
c:mm(a,b) -- store the result of a*b in c
```

What is torch ?

- Similar to Matlab / Python+Numpy







What is torch ?

- Easy integration into and from C
- Example: using CuDNN functions

```
for g = 0, self.groups - 1 do
    errcheck('cudnnConvolutionForward', cudnn.getHandle(),
             one:data(),
             self.iDesc[0], input:data() + g*self.input_offset,
             self.weightDesc[0], self.weight:data() + g*self.weight_offset,
             self.convDesc[0], self.fwdAlgType[0],
             self.extraBuffer:data(), self.extraBufferSizeInBytes,
             zero:data(),
             self.oDesc[0], self.output:data() + g*self.output_offset);
end
```

What is torch ?

- Strong GPU support

CUDA Tensors

Tensors can be moved onto GPU using the :cuda function

In [ ]:
```
require 'cutorch';
a = a:cuda()
b = b:cuda()
c = c:cuda()
c:mm(a,b) -- done on GPU
```

torch

Neural Networks

- nn: neural networks made easy
- building blocks of differentiable modules

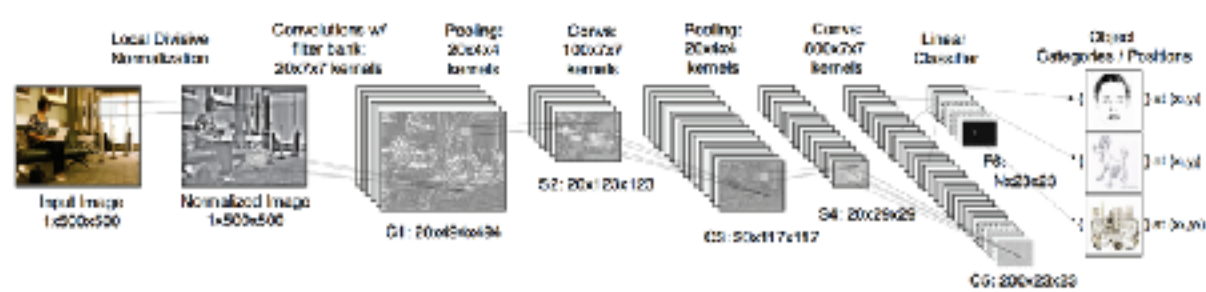
➡ **define a model with pre-normalization, to work on raw RGB images:**

```
model = nn.Sequential()

model:add( nn.SpatialConvolution(3,16,5,5) )
model:add( nn.Tanh() )
model:add( nn.SpatialMaxPooling(2,2,2,2) )
model:add( nn.SpatialContrastiveNormalization(16, image.gaussian(3)) )

model:add( nn.SpatialConvolution(16,64,5,5) )
model:add( nn.Tanh() )
model:add( nn.SpatialMaxPooling(2,2,2,2) )
model:add( nn.SpatialContrastiveNormalization(64, image.gaussian(3)) )

model:add( nn.SpatialConvolution(64,256,5,5) )
model:add( nn.Tanh() )
model:add( nn.Reshape(256) )
model:add( nn.Linear(256,10) )
model:add( nn.LogSoftMax() )
```



autograd by

- Write imperative programs
- Backprop defined for every operation in the language

```
neuralNet = function(params, x, y)
   local h1 = t.tanh(x * params.W[1] + params.b[1])
   local h2 = t.tanh(h1 * params.W[2] + params.b[2])
   local yHat = h2 - t.log(t.sum(t.exp(h2)))
   local loss = - t.sum(t.cmul(yHat, y))
   return loss
end

-- gradients:
dneuralNet = grad(neuralNet)

-- some data:
x = t.randn(1,100)
y = t.Tensor(1,10):zero() y[1][3] = 1

-- compute loss and gradients wrt all parameters in params:
dparams, loss = dneuralNet(params, x, y)
```

torch

Distributed Learning

- Multi-GPU and Multi-Node
- distlearn by Twitter
  - MPI-like model
  - scales to a large amount of nodes

```lua
-- Use a tau of 10 and an alpha of 0.2
local allReduceEA = require 'distlearn.AllReduceEA'(tree, 10, 0.2)
-- Make sure all the nodes start with the same parameter values
allReduceEA.synchronizeParameters(params)
for _ = 1,epochs do
   for _ = 1,steps
      -- Compute your gradients as normal
      local grads = computeYourGrads(...)
      -- Do your SGD as normal
      SGD(params, grads)
      -- Average the params
      allReduceEA.averageParameters(params)
   end
   -- Make sure the center's haven't drifted too far due to
   -- floating point precision error build up
   allReduceEA.synchronizeCenter(params)
   -- Validate...
end
```

Core Philosophy

- Interactive computing
    - No compilation time
- Imperative programming
    - Easy to understand, think and debug
- Minimal abstraction
    - Thinking linearly
- Maximal Flexibility
    - No constraints on interfaces or classes

torch

torch Community

NVIDIA

Facebook AI Research

UNIVERSITY OF OXFORD

École des Ponts ParisTech

NYU

canary

inria

IIIT-H

idiap RESEARCH INSTITUTE

VisionLabs visual recognition company

PURDUE UNIVERSITY

Yandex

element

TERADEEP

AMD

IBM

MULTICOREWARE

Moodstocks

torch Community

szagoruyko / **loadcaffe**

Watch 19 | Unstar 203 | Fork 59

Code | Issues 10 | Pull requests 1 | Wiki | Pulse | Graphs

Load Caffe networks in Torch7

facebook / **fb.resnet.torch**

Unwatch 62 | Unstar 421 | Fork 85

Code | Issues 4 | Pull requests 0 | Wiki | Pulse | Graphs

Torch implementation of ResNet from http://arxiv.org/abs/1512.03385 and training scripts

Moodstocks / **inception-v3.torch**

Watch 11 | Unstar 48 | Fork 8

Code | Issues 1 | Pull requests 0 | Wiki | Pulse | Graphs

Rethinking the Inception Architecture for Computer Vision http://arxiv.org/abs/1512.00567

torch Community

karpathy / neuraltalk2

Watch · Unstar · Fork

Code · Issues 45 · Pull requests 4 · Wiki · Pulse · Graphs

Efficient Image Captioning code in Torch, runs on GPU

NeuralTalk2

Recurrent Neural Network captions your images. Now much faster and better than the original NeuralTalk. Compared to the original NeuralTalk this implementation is **batched, uses Torch, runs on a GPU, and supports CNN finetuning**. All of these together result in quite a large increase in training speed for the Language Model (~100x), but overall not as much because we also have to forward a VGGNet. However, overall very good models can be trained in 2-3 days, and they show a much better performance.

This is an early code release that works great but is slightly hastily released and probably requires some code reading of inline comments (which I tried to be quite good with in general). I will be improving it over time but wanted to push the code out there because I promised it to too many people.

This current code (and the pretrained model) gets ~0.9 CIDEr, which would place it around spot #8 on the codalab leaderboard. I will submit the actual result soon.

a man is playing tennis on a tennis court

a train is traveling down the tracks at a train station

a cake with a slice cut out of it

a bench sitting on a patch of grass next to a sidewalk

You can find a few more example results on the demo page. These results will improve a bit more once the last few bells and whistles are in place (e.g. beam search, ensembling, reranking).

There's also a fun video by @kcimc, where he runs a neuraltalk2 pretrained model in real time on his laptop during a walk in Amsterdam.

torch

torch Community

jcjohnson / **neural-style**

Watch 389 | Unstar 7,152 | Fork 901

Code | Issues 56 | Pull requests 12 | Wiki | Pulse | Graphs

Torch implementation of neural style algorithm

neural-style

This is a torch implementation of the paper A Neural Algorithm of Artistic Style by Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge.

The paper presents an algorithm for combining the content of one image with the style of another image using convolutional neural networks. Here's an example that maps the artistic style of The Starry Night onto a night-time photograph of the Stanford campus:

torch Community

OpenNMT / OpenNMT

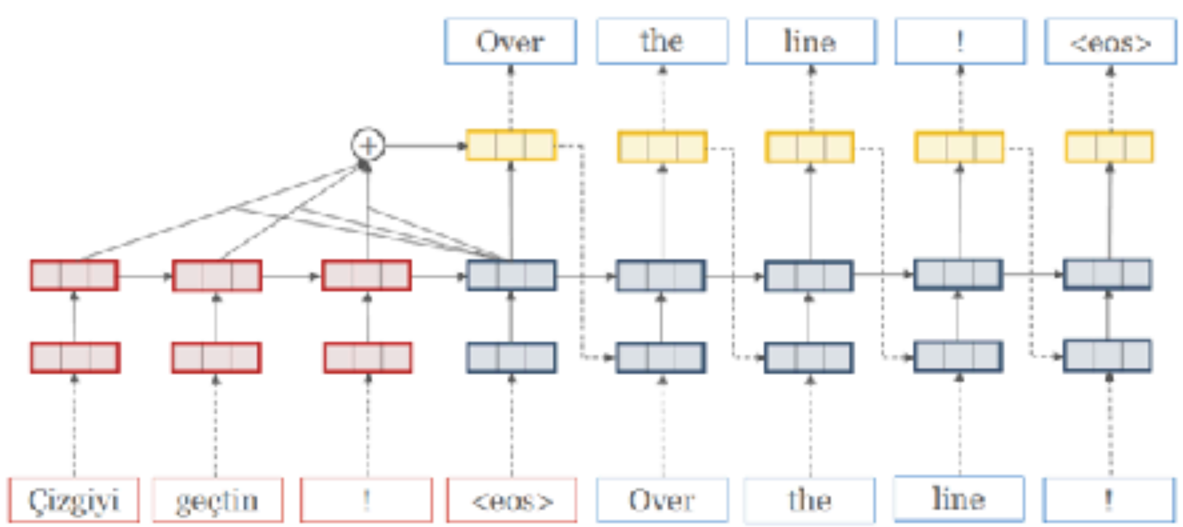
Watch 64 | Unstar 601 | Fork 121

Code | Issues 8 | Pull requests 4 | Projects 2 | Wiki | Pulse | Graphs

Open-Source Neural Machine Translation in Torch http://opennmt.net/

OpenNMT: Open-Source Neural Machine Translation

OpenNMT is a full-featured, open-source (MIT) neural machine translation system utilizing the Torch mathematical toolkit.

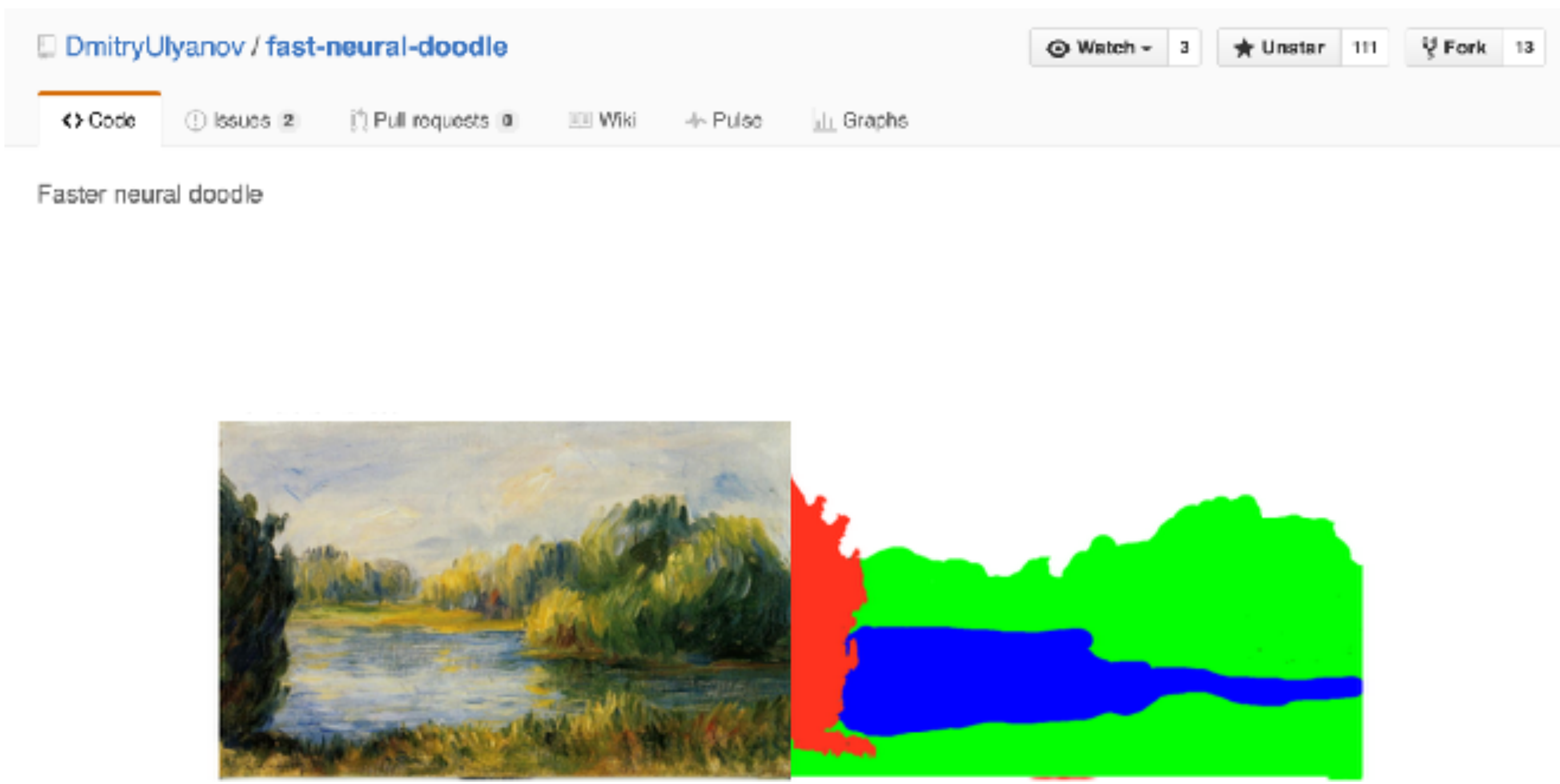


The system is designed to be simple to use and easy to extend , while maintaining efficiency and state-of-the-art translation accuracy. Features include:

- Speed and memory optimizations for high-performance GPU training.
- Simple general-purpose interface, only requires and source/target data files.
- C++ implementation of the translator for easy deployment.
- Extensions to allow other sequence generation tasks such as summarization and image captioning.

torch

torch Community



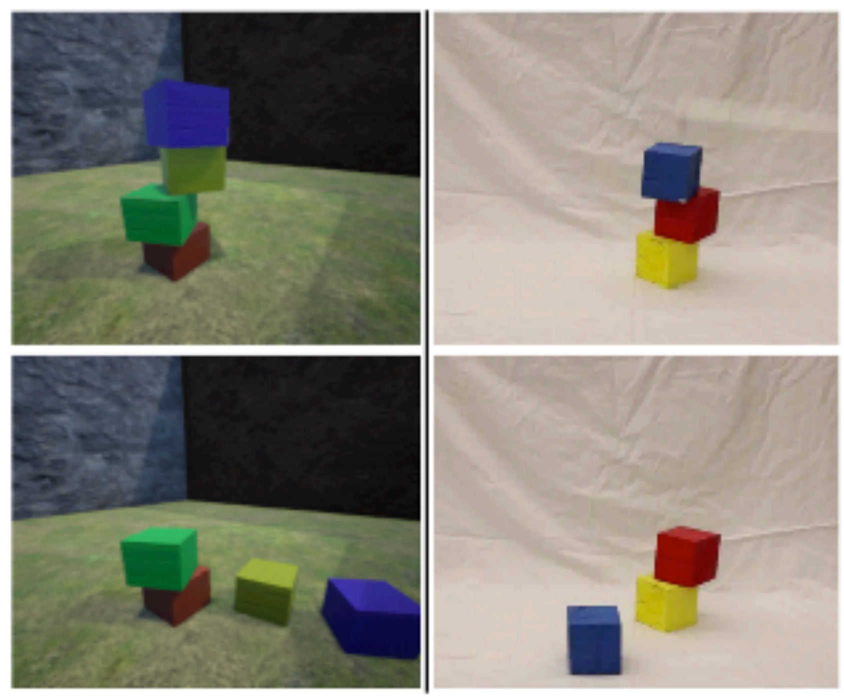
torch Community

facebook / UETorch

Unwatch 37 | Unstar 135 | Fork 20

Code | Issues 1 | Pull requests 0 | Wiki | Pulse | Graphs

A Torch plugin for Unreal Engine 4.



TorchCraft / TorchCraft

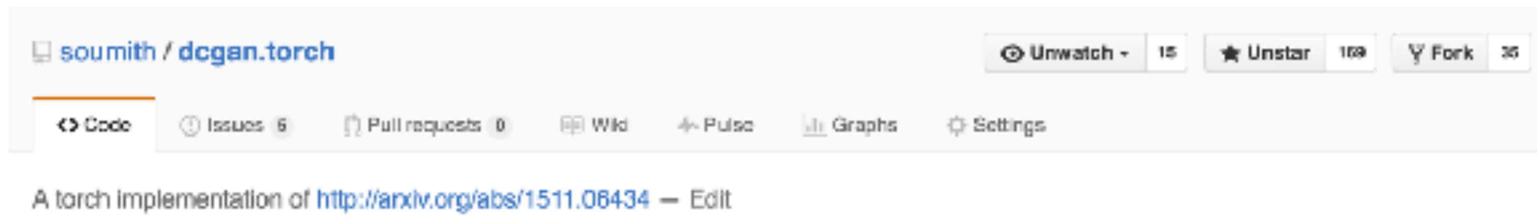
Unwatch 43 | Unstar 608 | Fork 77

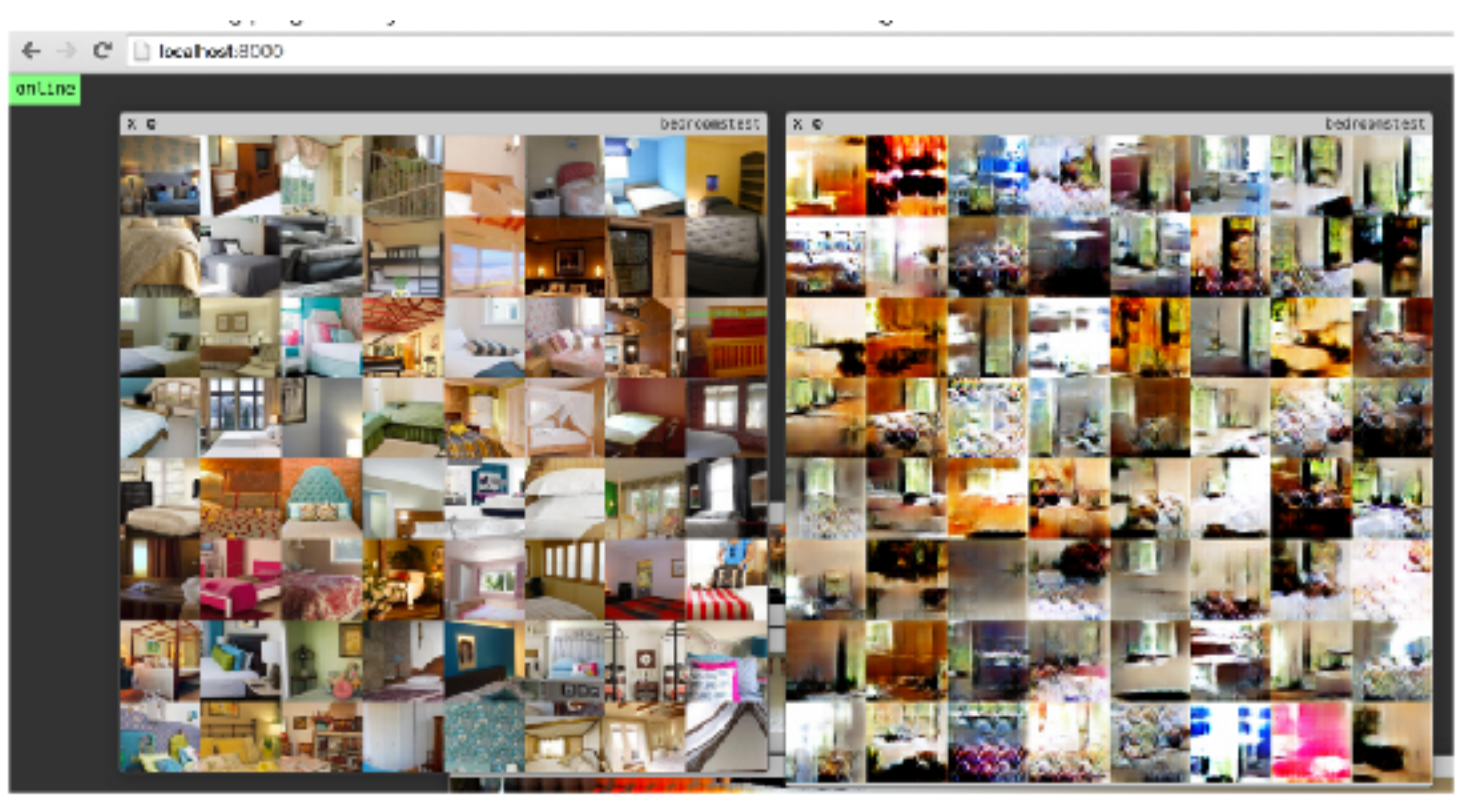
Code | Issues 4 | Pull requests 0 | Projects 0 | Wiki | Pulse | Graphs | Settings

Connecting Torch to StarCraft

Edit

torch Community





Today's AI

- Reasonably solved to be used in production

Today’s AI

- Classification and Detection
- in images, videos, volumetric data, sparse datasets

DenseCap: Johnson et. al. https://github.com/jcjohnson/densecap

Today's AI

- Segmentation

DeepMask: Pinhero et. al.

torch

Today's AI

- Text Classification (sentiment analysis etc.)
- Text Embeddings
- Graph embeddings
- Machine Translation
- Ads ranking

Today’s AI

static datasets + static model structure

Today's AI

static datasets + static model structure

model does not change over training time

Today's AI

static datasets + static model structure

model does not change over training time

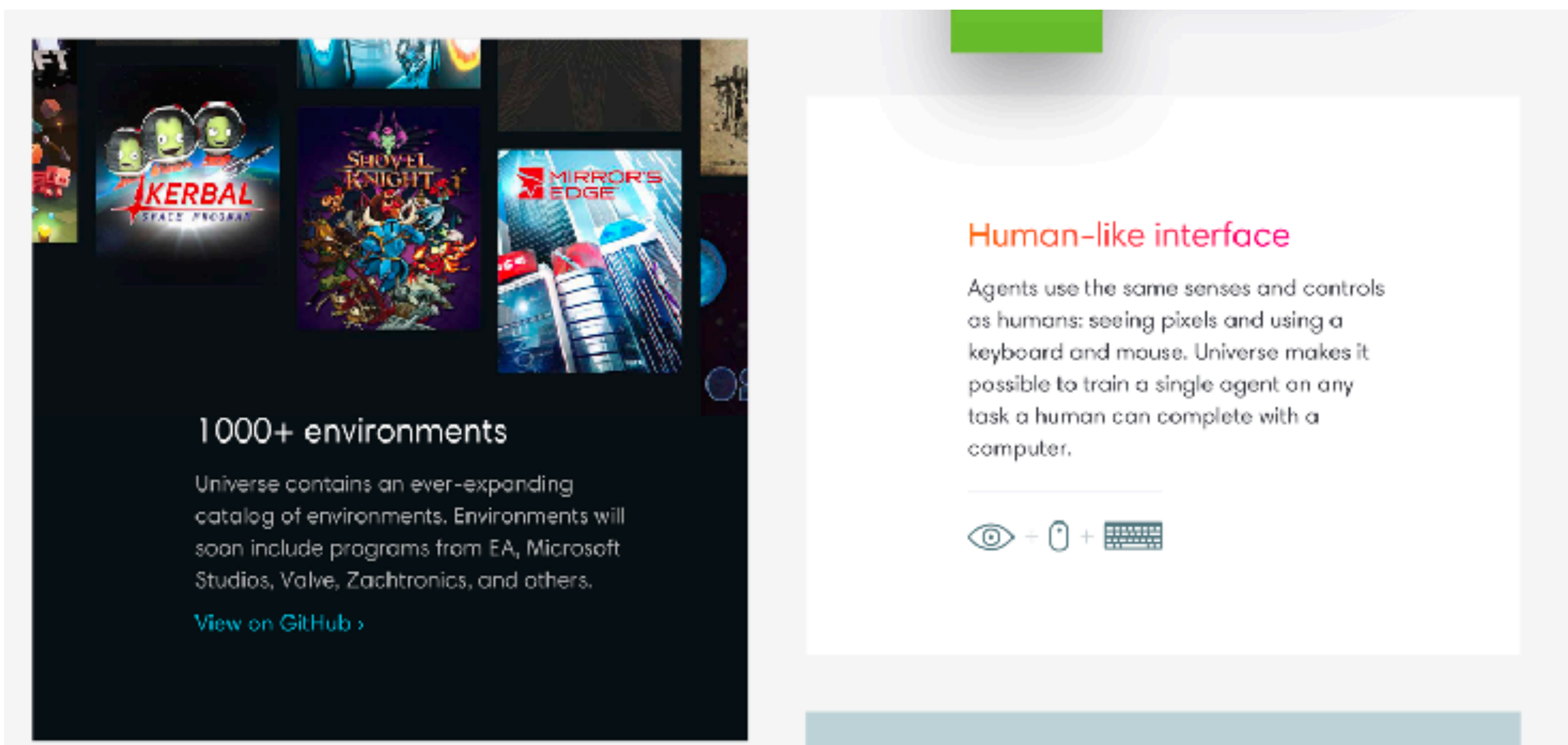
offline learning

Active and Future AI Research

- Agents training in different and new environments

Active and Future AI Research

- RL Agents in different and new environments
  - Example: OpenAI Universe



torch

Active and Future AI Research

- RL Agents in different and new environments
  - Example: OpenAI Universe
- Online Learning
  - Example: self-driving cars

Active and Future AI Research

- RL Agents in different and new environments
  - Example: OpenAI Universe
- Online Learning
  - Example: self-driving cars
- Dynamic Neural Networks
  - self-adding new memory or layers
  - changing evaluation path based on inputs

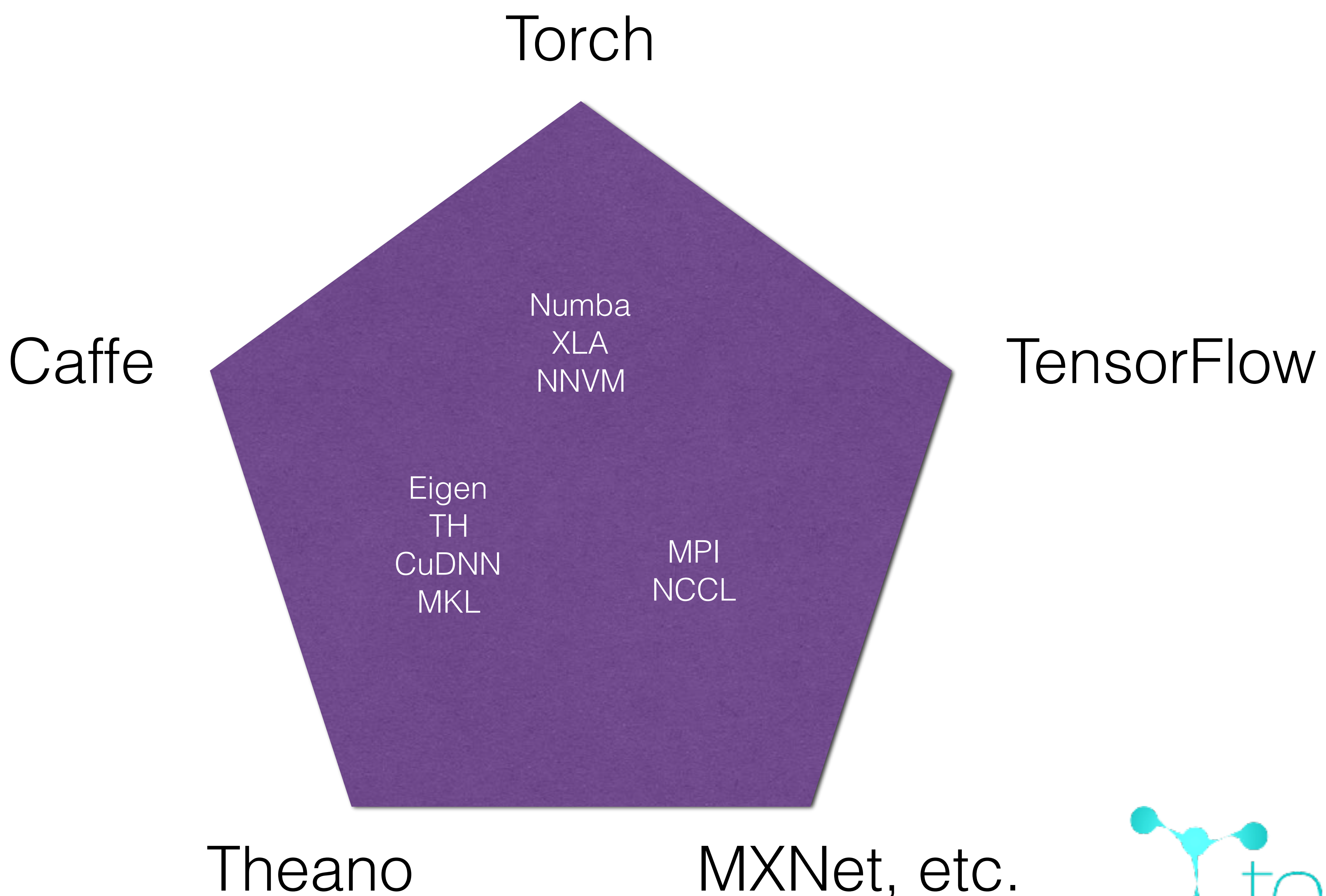
Active and Future AI Research

- RL Agents in different and new environments
  - Example: OpenAI Universe
- Online Learning
  - Example: self-driving cars
- Dynamic Neural Networks
  - self-adding new memory or layers
  - changing evaluation path based on inputs
- Structured Prediction
  - Viterbi style decoders

A Next-Gen Framework

- Interop with environments
  - Like Universe, VizDoom, etc.
  - Easy data loaders
- Rich scientific ecosystem (such as SciPy)
  - Neural Networks are not the only methods to be used
- Dynamic Neural Networks
  - with no specific or static structure
- Minimal Abstractions
  - better debugging and low-level programming in the hands of researcher
- Graph Compilers
  - To speed up structured prediction and fuse operations
  - Numba, XLA, Theano

Let’s share!


Torch
Caffe
TensorFlow
Theano
MXNet, etc.
Numba
XLA
NNVM
Eigen
TH
CuDNN
MKL
MPI
NCCL