Advanced Machine Learning 2014
(Introduction to Torch)

Jost Tobias Springenberg, Jan Wülfing
Martin Riedmiller
{springj,wuelfj,riedmiller}@cs.uni-freiburg.de

Machine Learning Lab
University of Freiburg

November 25, 2014

Today...

- **Torch:** I will give a short (hopefully concise) introduction to torch

Today...

- **Torch:** I will give a short (hopefully concise) introduction to torch
- **Assignment:** We will give out the second assignment training a MLP / ConvNet to recognize digits using torch + testing on a secret test-set ;)

Today...

- **Torch:** I will give a short (hopefully concise) introduction to torch
- **Assignment:** We will give out the second assignment training a MLP / ConvNet to recognize digits using torch + testing on a secret test-set ;)
  - Part 1: Reproduce results (maybe even state of the art) from previous work

Today...

- **Torch:** I will give a short (hopefully concise) introduction to torch
- **Assignment:** We will give out the second assignment training a MLP / ConvNet to recognize digits using torch + testing on a secret test-set ;)
  - Part 1: Reproduce results (maybe even state of the art) from previous work
  - Part 2: Train the best network you can design for MNIST and report classification error

Today...

- **Torch:** I will give a short (hopefully concise) introduction to torch
- **Assignment:** We will give out the second assignment training a MLP / ConvNet to recognize digits using torch + testing on a secret test-set ;)
  - Part 1: Reproduce results (maybe even state of the art) from previous work
  - Part 2: Train the best network you can design for MNIST and report classification error
  - Part 3: Give us up to 3 models that we can call using lua $\rightarrow$ we will test this on our secret test-data

Today...

- **Torch:** I will give a short (hopefully concise) introduction to torch
- **Assignment:** We will give out the second assignment training a MLP / ConvNet to recognize digits using torch + testing on a secret test-set ;)
  - Part 1: Reproduce results (maybe even state of the art) from previous work
  - Part 2: Train the best network you can design for MNIST and report classification error
  - Part 3: Give us up to 3 models that we can call using lua $\rightarrow$ we will test this on our secret test-data
- **Regarding the presentation:** As before I have borrowed slides/examples from the torch developers who do a much better job than me at explaining their work

Popular (Deep) Machine Learning Frameworks

- **Scikit-learn (python)** a widely used machine learning framework in python building on numpy

Popular (Deep) Machine Learning Frameworks

- **Scikit-learn (python)** a widely used machine learning framework in python building on numpy
  - → large amount of different algorithms for all learning scenarios (supervised/unsupervised)
  - → some wrappers for neural networks
  - → good for rapid prototyping
  - → not (yet) so great for large scale neural networks

Popular (Deep) Machine Learning Frameworks

- **Scikit-learn (python)** a widely used machine learning framework in python building on numpy
  - → large amount of different algorithms for all learning scenarios (supervised/unsupervised)
  - → some wrappers for neural networks
  - → good for rapid prototyping
  - → not (yet) so great for large scale neural networks
- **Theano (python)** a python language extension that allows you to define/evaluate mathematical expressions by compiling them to CPU / GPU code

Popular (Deep) Machine Learning Frameworks

- **Scikit-learn (python)** a widely used machine learning framework in python building on numpy
  - → large amount of different algorithms for all learning scenarios (supervised/unsupervised)
  - → some wrappers for neural networks
  - → good for rapid prototyping
  - → not (yet) so great for large scale neural networks
- **Theano (python)** a python language extension that allows you to define/evaluate mathematical expressions by compiling them to CPU / GPU code
  - → re-creates most functionality of numpy (and plays well with it together)
  - → compiles to quite fast code!
  - → computes gradients automatically!
  - → has an amazing neural network toolbox built on top (pylearn2)
  - → can be hard to understand/extend/debug for beginners (you define a graph of functions rather than writing code directly)

Popular (Deep) Machine Learning Frameworks

- **Caffe (c++)** a fast and readable implementation of ConvNets in c++

Popular (Deep) Machine Learning Frameworks

- **Caffe (c++)** a fast and readable implementation of ConvNets in c++
  - → extremely fast and well written c++ code
  - → pre-trained models for ImageNet etc.
  - → less good for rapid prototyping
  - → restricted to (convolutional) neural networks

Popular (Deep) Machine Learning Frameworks

- **Caffe (c++)** a fast and readable implementation of ConvNets in c++
  - → extremely fast and well written c++ code
  - → pre-trained models for ImageNet etc.
  - → less good for rapid prototyping
  - → restricted to (convolutional) neural networks
- **Torch (lua)** a Matlab-like Environment for (deep and not so deep) Machine Learning

Popular (Deep) Machine Learning Frameworks

- **Caffe (c++)** a fast and readable implementation of ConvNets in c++
  - → extremely fast and well written c++ code
  - → pre-trained models for ImageNet etc.
  - → less good for rapid prototyping
  - → restricted to (convolutional) neural networks
- **Torch (lua)** a Matlab-like Environment for (deep and not so deep) Machine Learning
  - → comes with an extremely versatile tensor (matrix) implementation
  - → is very fast and easily packageable (thanks to lua)
  - → computes gradients (semi)-automatically
  - → provides high-level language for rapid-prototyping (easily understandable)
  - → provides multiple tensor backends for even faster GPU/CPU computation
  - → requires you to learn lua
  - → (arguably) has less pre-trained state of the art models

Torch: Getting Started

Your quick start into torch land:

- Torch's **main site** and resources: www.torch.ch
  On Github: https://github.com/torch
- Torch **cheat sheet**
  https://github.com/torch/torch7/wiki/Cheatsheet
- **Tutorials** for Torch: http://torch.madbits.com
  On Github: https://github.com/clementfarabet/torch-tutorials
- Lua: http://www.lua.org
- LuaJIT: http://luajit.org/luajit.html
- Extensions by research teams of Google,facebook and wrappers to libraries by nvidia (see end of presentation)

Torch: Why Lua

Why does torch use lua with C++ backends ?

- Just programming C can be very tiresome for research code
  - → proper scripting language (LuaJIT)
- However C speed (and beyond) needed for large scale applications
  - → compiled and optimized backend (C,C++,CUDA,OpenMP)

Torch: Why Lua

Why does torch use lua with C++ backends ?

- Just programming C can be very tiresome for research code
  - → proper scripting language (LuaJIT)
- However C speed (and beyond) needed for large scale applications
  - → compiled and optimized backend (C,C++,CUDA,OpenMP)

Why not python ?
→ Luajit provides

Torch: Why Lua

Why does torch use lua with C++ backends ?

- Just programming C can be very tiresome for research code
  → proper scripting language (LuaJIT)
- However C speed (and beyond) needed for large scale applications
  → compiled and optimized backend (C,C++,CUDA,OpenMP)

Why not python ?
→ Luajit provides

- **Fast**est scripting language, with a transparent JIT compiler
- **Simple**, readable (like Python)
- The **clean**est interface to C (although now there is julia)
- **Embeddable** into any environment (iPhone apps, Video games, web backends ...)

Lua: a very short introduction

- A longer version (15 minutes) http://tylerneylon.com/a/learn-lua/
- The real deal: http://luatut.com/, http://www.lua.org/docs.html
- Good cheat sheets: http://thomaslauer.com/download/luarefv51.pdf

Let us look at code !

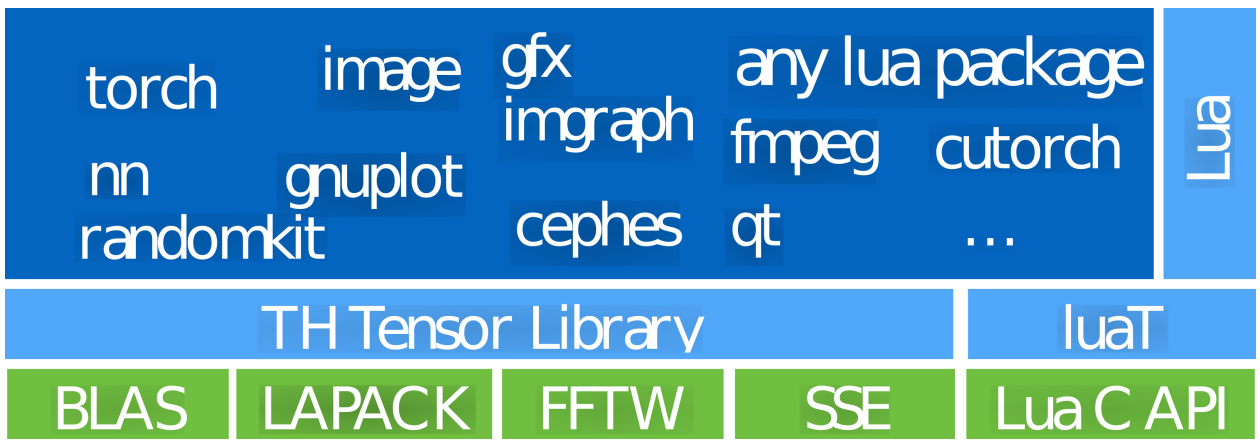
Torch: another very short introduction

- **Torch7** extends lua with one more object: the **Tensor**
- It is similar to matlab arrays or numpy arrays

Let us look at more code !

Torch: ecosystem

- **Torch7** also provides a large set of packages
  - based on Matlab's common routines (zeros,ones,eye, ...)
  - Linear algebra routines
  - Convolutions, Fourier transform, etc.



Torch: ecosystem

Package Manager

→ more packages are available via Lua's package manger: **luarocks**

- check out what's available here: http://github.com/torch/rocks

Torch: Neural Networks

The **nn** package

- When training neural nets, linear regression, convolutional networks; basically any machine learning model, we're interested in gradients, and loss functions
- The nn package provides a large set of transfer functions, which all come with three methods:
  - upgradeOutput() – compute the output given the input
  - upgradeGradInput() – compute the derivative of the loss wrt input
  - accGradParameters() – compute the derivative of the loss wrt weights
- The nn package provides a set of common loss functions, which all come with two methods:
  - upgradeOutput() – compute the output given the input
  - upgradeGradInput() – compute the derivative of the loss wrt input
- You don't have to compute your gradients yourself anymore ! :)
- There exist **optimized** backends for almost all modules in **nn**

Torch: Neural Networks

Let's look at how to solve your exercise from Phase 1

Torch: Neural Networks

- Arbitrary models can be constructed using LEGO-like containers:

nn.Sequential() – sequential modules
nn.ParallelTable() – parallel modules
nn.ConcatTable() – shared modules
nn.SplitTable() – (N)dim Tensor, table of (N-1)dim Tensors
nn.JoinTable() – table of (N-1)dim Tensors, (N)dim Tensor

Torch7 Resources Summary:

- → Torch7:
  http://www.torch.ch/ https://github.com/torch
- → Basic Demos: a bunch of demos/tutorials to get started
  https://github.com/clementfarabet/torch7-demos
- → Deep-Learning Tutorials: supervised and unsupervised learning
  http://code.madbits.com
- → **luarocks**: Lua's package manager, to get new packages:
  luarocks search –all # list all packages
  luarocks install optim # install optim package
- → Torch Group: get help!
  https://groups.google.com/forum/?fromgroups#!forum/torch7
  - For **benchmarks comparing caffe/theano/torch/cuda-convet**
    https://github.com/soumith/convnet-benchmarks
- → Useful packages from facebook:
  https://github.com/facebook/fblualib
  python bindings, serialization, better repl