Language Modeling with Torch

Nicholas Leonard

Element Inc.

https://github.com/nicholas-leonard

torch

Agenda

- Meeting Torch
  - What is Torch?
  - When should I use Torch?
  - How does Torch work?
- Language Modeling with Torch
  - Understanding Language Models
  - Recurrent Neural Networks
  - Generating sentences

torch

Meeting Torch

Lua, tensors and neural networks

What is Torch?
Lua, not Python

- Examples of Lua types :

String

```
a = 'hello'
string.sub(a, 'he[l]+)', 'jell')
print(a)
jello
```

Number

```
b = 3
print(b + 4)
7
```

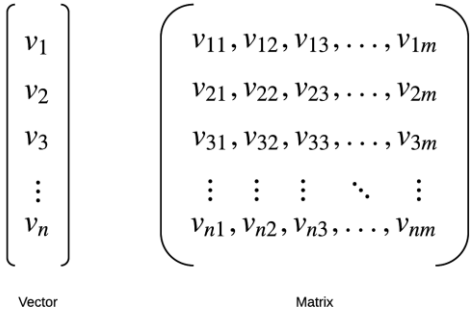
Table

```
c = {1,2,3,key='value'}
c.key = 4
c[2]= 7
print(c)
{1,7,3,key=4}
```

- Lua is lightweight, elegant and easy to learn
- Easy to interface with C/CUDA
- Fast for-loops
- Functions as first class citizens, closures

torch

What is Torch?
Tensors for linear algebra



- **Tensors are N-dimension arrays**
  - **Initialize and matrix-matrix multiply two tensors:**

```
a = torch.Tensor(2,3)
a:uniform(0,1)
print(a)
 0.6187 0.8752 0.1496
 0.6979 0.8230 0.1439
[torch.DoubleTensor of size 2x3]
```

```
b = torch.Tensor{{1},{2},{3}}
print(b)
 1
 2
 3
[torch.DoubleTensor of size 3x1]
```

```
c = torch.mm(a,b)
c:mm(a,b)
c = a * b
```

- **Basic linear algebra sub-routines (BLAS)**
- **Sub-tensor extraction, etc.**
- **Can be run on GPU/CPU**
- **Different types : DoubleTensor, FloatTensor, LongTensor, CudaTensor, etc.**

torch

What is Torch?
Neural Networks for deep learning

- Train neural networks using back-propagation (gradient descent + chain rule)

```
-- model
mlp = nn.Sequential()
   :add(nn.Linear(3,4))
   :add(nn.Tanh())
   :add(nn.Linear(4,2))
   :add(nn.Sigmoid())
-- loss function
mse = nn.MSECriterion()
```

```
-- data sample
input = torch.randn(3)
target = torch.randn(2)
```

```
-- forward
output = mlp:forward(input)
loss = mse:forward(output, target)
-- backward
gradOutput = mse:backward(output, target)
mlp:zeroGradParameters()
mlp:backward(input, gradOutput)
-- update
mlp:updateParameters(0.1)
```

- Automatic gradient differentiation
  - No compilation (TensorFlow, Theano)
  - Imperative programming (easier to debug than symbolic graphs)
- Extend by implementing nn.Modules or nn.Criterions

torch

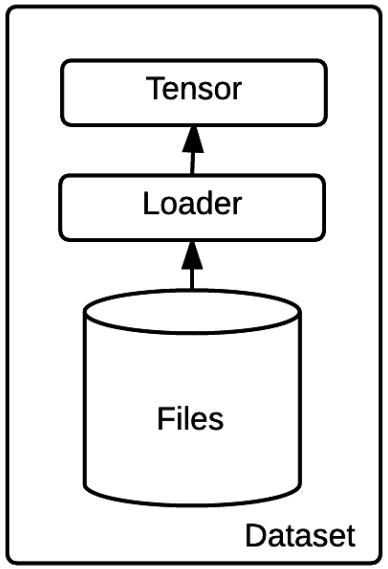
When should Torch be used? Research and production

- Deep learning research
  - Assemble existing Modules and Criterions
  - Or implement your own
  - Easy to debug (no compilation or symbolic graph)
  - Lots of packages to pick from (torch, nn, optim, image, dataload, rnn, etc.)
- Production environment
  - Lua was designed for embedded systems
  - Servers (GPU/CPU, Ubuntu/Linux, Max OS X)
  - Smart phones (Android, iOS)
- Fast execution
  - glample/rnn-benchmarks
  - soumith/convnet-benchmarks

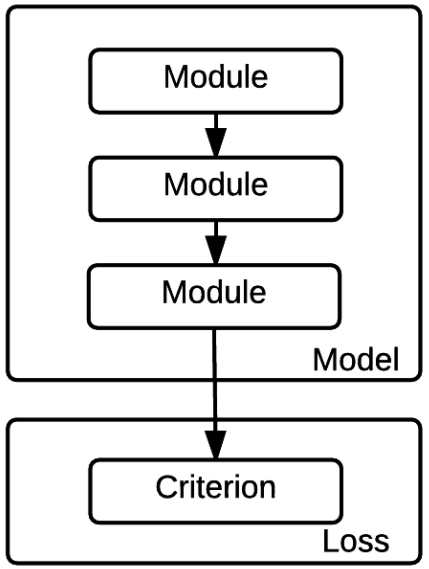
torch

How does Torch work?
Mental model

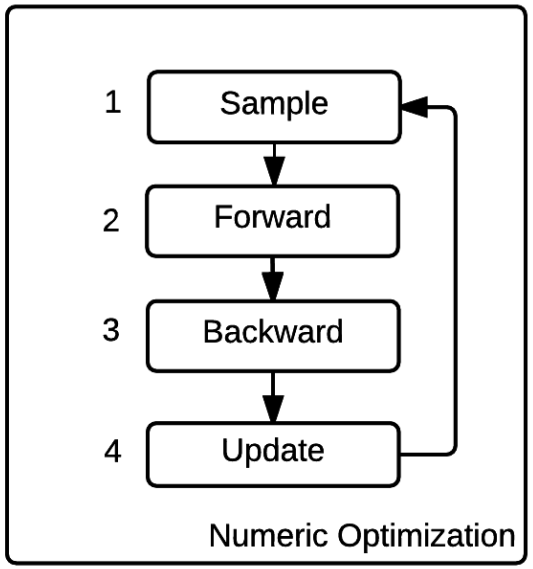
Prepare Dataset



Define Model and Loss



Train Model



torch

Language Models

Generating sentences using recurrent neural networks

torch

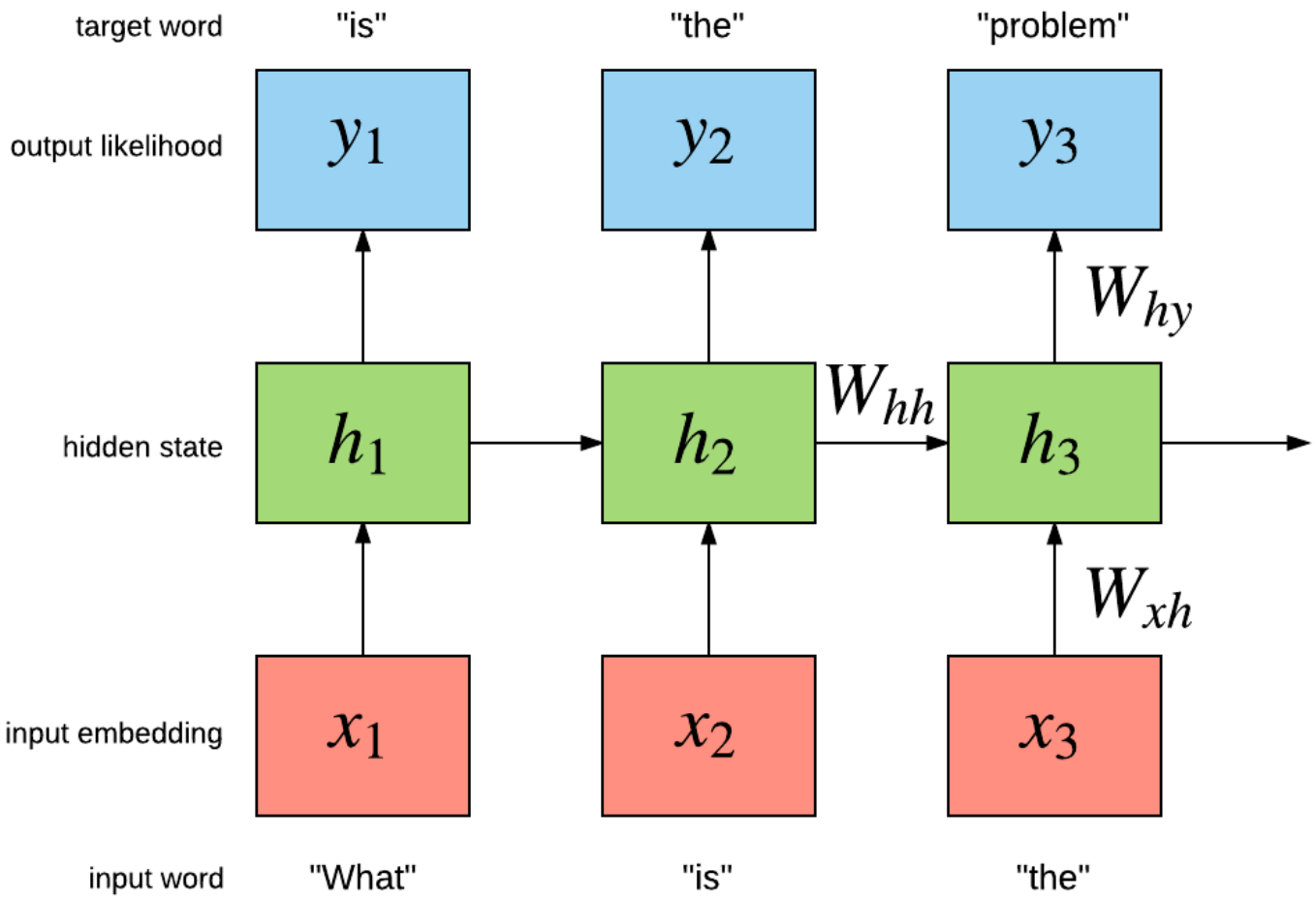
Understanding language models

- Language model dataset is a corpus of text samples
  - “Why does it always rain on Saturdays?”
- Maximize the likelihood of a word given previous words: $P(w_t|w_1, ..., w_{t-1})$
- Applications:
  - Generate sentences (sample next word given previous words)
  - Auto-complete
  - Measure the probability of a sentence (e.g. does this translation make sense?)

$$P(w_1, w_2, ..., w_T) = \prod_{t=1}^{T} P(w_t|w_1, ..., w_{t-1})$$

torch

Recurrent Neural Networks



torch

Recurrent Neural Networks

- Model architectures
  - Simple Recurrent Neural Networks (Simple RNN)
  - Long Short Term Memory (LSTM)
  - Gated Recurrent Units (GRU)
- Used to model sequential data
  - Video frames, text, time-series
- Different applications
  - Language modeling, sentiment analysis, machine translation
- Element-Research/rnn package provides RNNs for Torch
  - Tutorials, documentation, models, criterions and examples

torch

Generating sentences

- **Recursively sample sentences (one word at a time)**

<S> The company said its net profit rose to $ 289 million , or 96 cents per share , in the three months ended on March 31 compared with $ 173 million , or $ 0.68 a share , a year ago . </S>
<S> But I 've been a bit disappointed with our performance , " said Wenger . </S>
<S> The first is an even bigger problem . </S>
<S> The next big thing for him is he will be able to tell the world he is thinking about his future . </S>
<S> The new rules have been added to the legislation so that they don 't have to be approved for public use . </S>
<S> The Pentagon 's top counter-terrorism official , who has been in charge of a new system of intelligence collection and inspection , wrote in an e-mail message that while the new system could be easily implemented , it remains an option . </S>
<S> " I was trying to get a glass of water . </S>
<S> Later he was driven to a nearby house where he was later found to be severely ill . </S>

- **40GB model learns to generate coherent sentences**
- **Trained on Google 1-billion words dataset**

torch

Multi-GPU Multi-Layer LSTM

```
nn.Serial @ nn.Sequential {
  [input -> (1) -> (2) -> (3) -> output]
  (1): nn.ParallelTable {
    input
      |`-> (1): nn.Sequential {
      |      [input -> (1) -> (2) -> (3) -> (4) -> (5) -> (6) -> (7) -> (8) -> (9) -> (10) -> (11) -> (12) -> output]
      |      (1): nn.Convert
      |      (2): nn.GPU(2) @ nn.Concat {
      |        input
      |          |`-> (1): nn.GPU(1) @ nn.LookupTableMaskZero
      |          |`-> (2): nn.GPU(2) @ nn.LookupTableMaskZero
      |           ... -> output
      |      }
      |      (3): nn.GPU(2) @ nn.Dropout(0.2, busy)
      |      (4): nn.GPU(1) @ nn.SeqLSTM
      |      (5): nn.GPU(1) @ nn.Dropout(0.2, busy)
      |      (6): nn.GPU(1) @ nn.SeqLSTM
      |      (7): nn.GPU(1) @ nn.Dropout(0.2, busy)
      |      (8): nn.GPU(2) @ nn.SeqLSTM
      |      (9): nn.GPU(2) @ nn.Dropout(0.2, busy)
      |      (10): nn.GPU(2) @ nn.SeqLSTM
      |      (11): nn.GPU(2) @ nn.Dropout(0.2, busy)
      |      (12): nn.GPU(3) @ nn.SplitTable
      |    }
      |`-> (2): nn.Identity
       ... -> output
  }
  (2): nn.ZipTable
  (3): nn.GPU(3) @ nn.Sequencer @ nn.Recursor @ nn.MaskZero @ nn.NCEModule(2048 -> 793471)
}
```

torch

For more info:

http://torch.ch
https://github.com/Element-Research/rnn