An Introduction to Theano

Semih Yagcioglu
Deep Learning

What society thinks I do

What my friends think I do

What other computer scientists think I do

What mathematicians think I do

What I think I do

What I actually do

from theano import *
What is Theano?
Theano is many things

- A mathematical symbolic expression compiler
- A Python library for symbolic maths - far broader than just Deep Learning
- Tightly integrated with the Python ecosystem
- Fast C/CUDA back-end and transparent GPU acceleration
What Theano is not

• Theano is not a machine learning toolkit
• It’s a mathematical toolkit that makes building downstream machine learning models easier.
Project Status

- Theano has been developed and used since 2008, by LISA lab at the University of Montreal (led by Yoshua Bengio)
- Several Deep Learning Tutorials are based on Theano
- Several Machine learning libraries built upon Theano  
  - Blocks, Pylearn2, Keras, Lasagne, etc.
- Good user documentation  
  - http://deeplearning.net/software/theano/
- Open-source on Github
Basics of Theano
Symbolic Expression Compiler

Variables and expressions are symbolic - more like maths than code...

```python
import theano
from theano import tensor as T

x = T.vector('x')
W = T.matrix('W')
b = T.vector('b')

dot = T.dot(x, W)
out = T.nnet.sigmoid(dot + b)
```

...but, symbolic expressions use a familiar NumPy-like syntax
How Does It Work?

Theano defines a **language**, a **compiler** and a **library**

Recipe for a Theano application:

- Define symbolic expressions
- Compile a function that can compute numeric values using those expressions
- Execute that function on data
Tensor

- Tensor: multi-dimensional array
  - Order of tensor: dimensionality
    - $0^{th}$-order tensor = scalar
    - $1^{th}$-order tensor = vector
    - $2^{th}$-order tensor = matrix
    - ...

Scalar Math

from theano import tensor as T

x = T.scalar()
y = T.scalar()
z = x + y
w = z * x
a = T.sqrt(w)
b = T.exp(a)
c = a ** b
d = T.log(c)
Vector Math

from theano import tensor as T

x = T.vector()
y = T.vector()
# Scalar math applied elementwise
a = x * y
# vector dot product
b = T.dot(x, y)
from theano import tensor as T

x = T.matrix()
y = T.matrix()
a = T.vector()
# Matrix-matrix product
b = T.dot(x, y)
# Matrix-vector product
c = T.dot(x, a)
Putting Things Together

\[ y = a \times b \]

\[ a, b \in \mathbb{R} \]
Putting Thing Together In Theano

```
import theano
from theano import tensor as T

a = T.scalar()  # Initialize symbolic variables
b = T.scalar()

y = a * b  # Define symbolic expression

multiply = theano.function(inputs=[a, b], outputs=y)  # Compile a function

print multiply(3, 2)  # Use on numeric data
print multiply(4, 5)  # #20
```

Nvidia
How to install Theano
Requirements

- OS: Linux, Mac OS X, Windows
- Python: >= 2.6
- Numpy, Scipy, BLAS
Installing Stable Release

• It’s quite easy
  – pip install [--upgrade] theano
  – easy_install [--upgrade] theano
• But if you are to use a wrapper, better to use the latest version
Installing From Source

• It’s quite easy too
  – pip install --upgrade --no-deps git+git://github.com/Theano/Theano.git

• Yet it might cause problems
Did I Install It Correctly?

• You can easily test whether you installed it correctly or not by just importing it
  – import theano
Conclusions

- Theano is quite powerful and built with speed in mind
- Things are easier to implement thanks to Python
- Plenty of useful operations such as gradient calculation, cost computation etc.
- There are plenty of wrappers such as Keras and Lasagne