**THEANO – CHEAT SHEET**

**IMPORTS**

* import theano -main module
* from theano import tensor as T-tensor operations

**SYMBOLIC VARIABLES:**

* *X = T.matrix ()*
* Scalar, vector, matrix, tensor…..
* Types are specified by i/f/d

Operations with symbolic variables

* Basic operators like +,-,\*, /

Dot product*: T.dot (A, B) or A.dot (B)*

* Aggregations: T.min (), T.max (), T.mean (), T.sum (), etc.
* Reshaping: Use existing dimensions and use ‘X’ for new broadcastable dimension
* *A.reshape ((10, 1)), A.reshape ((10)), A.reshape ((2,5 )), etc.*
* *A.dimshuffle ((1, 0)), A.dimshuffle ((’x’, 0)), etc.*
* Subtendor operators: indexing as numpy and slicing similar to numpy
* *T.set\_subtensor (subtensor, value)*
* *T.inc\_subtensor (subtensor, inc )*

**SHARED VARIABLES**

* Variables having persistent values
* It is an ideal for model parameters
* *w = theano.shared (np.random.rand (100).astype(’float32’)*
* It is initialized with numpy arrays
* Optional name parameter
* Used as a variable in symbolic expressions
* *W.get\_value (), W.set\_value (new\_value)*
* Getting/setting value
* Use when it is necessary
* Borrows parameter

False: copy given array

True: reference given array

**COMPUTATIONAL GRAPH**

* Variable operations may produce other symbolic variables
* Computational graph

Node: symbolic expression resulting variable

Directed edges

In: variables need for computing

Out: to variables used in computation

**FUNCTIONS**

* Parts complied with a function
* *C = A + B*
* *add = theano.function ([A,B], C*
* Theano.function
* First parameter: list of inputs

Always list

All variables should be provided with variables

Shared variables should not be there

* Second parameter: outputs

Optional

* Updates: dictionary of updates

Calling function: res=add (1,1)

**GRADIENT**

* *T.grad(X, wrt=W)*
* Compute gradient x respect to w
* W must be a part of computation

**UPDATING SHARED VARIABLES**

* Update dictionary: contains both shared and symbolic variables with same shape, size…….
* *from collections import OrderedDict*
* *updates = OrderedDict ()*
* *updates [W] = 2\*W*
* dictionary is used in the update parameter of a function
* *mult\_by\_2 = theano.function ([], updates=updates)*
* shared variable will update function every time when the function is called on the basis of the rule defined by dictionary

**RANDOM NUMBER GENERATION**

* *from theano.sandbox.rng\_mrg import MRG\_RandomStreams as RandomStreams*
* *srng = RandomStreams ()*
* *srng.binomial ((10, 5), p=0.5, dtype=theano.config.floatX)*