Theano cheatsheet

(for detailed information on Theano see: http://deeplearning.net/software/theano)

Imports:

- `import theano` #main module
- `from theano import tensor as T` #variables & vector/matrix/tensor operations

Symbolic variables

- `X = T.matrix()`
- scalar, vector, matrix, tensor, etc.
- Type can be specified by using i/f/d for integer/float32/double: `X = T.ivector()`

Operations with symbolic variables

- Basic elementwise operations: +, -, *, /, **, etc.
- Dot product: `T.dot(A, B)` or `A.dot(B)`
- Aggregations: `T.min()`, `T.max()`, `T.mean()`, `T.sum()`, etc.
  - axis: only aggregate along the specified dimension
    - Maximum per column of matrix A: `T.max(A, axis=0)`
    - Sum of rows of matrix A: `T.sum(A, axis=1)`
- Reshaping
  - `A.reshape((10, 1))`, `A.reshape((10))`, `A.reshape((2,5))`, etc.
  - `A.dimshuffle((1,0))`, `A.dimshuffle(('x', 0))`, etc.
    - Use already existing dimension indeces and use 'x' for a new broadcastable dimension (e.g. ('x', 0) creates a matrix row (2D) from a 1D vector))
  - `A.T` #transpose
- Subtensor operators
  - Indexing as in numpy (Might be slow on GPU! Indexing 2D matrices by row is fast.)
  - Slicing similar to numpy
  - `T.set_subtensor(subtensor, value)`
  - `T.inc_subtensor(subtensor, inc)`

Shared variables

- Variables with persistent value
- Ideal for model parameters
- `w = theano.shared(np.random.rand(100).astype('float32'))`
  - Initialized with numpy arrays (determines shape, size, type and starting values)
    - In the example above: a vector of 100 float32 values
  - Optional name parameter
- Can be used as a variable in symbolic expressions
- Any operation with shared variables results in a symbolic experession/variable
- `W.get_value()`, `W.set_value(new_value)`
  - Getting/setting the value of a symbolic variable
  - Use only when necessary
  - borrow parameter