Torch for Matlab® Users

Ata Mahjoubfar
ata.m@ucla.edu

April 21, 2015

General

Get help for a specific function:

Matlab	`help sqrt`
Torch	`help(torch.sqrt)`

Matrices and Tensors

Create a two-dimensional tensor or matrix:

Matlab	`m = [9, 6, 3, 4; 7, 2, 8, 1]`
Torch	`m = torch.Tensor({{9, 6, 3, 4}, {7, 2, 8, 1}})`

Create a row vector:

Matlab	`v = [9, 7, 6, 8]` or `v = [9 7 6 8]`
Torch	`v = torch.Tensor({{9, 7, 6, 8}})`

But, this is not used as a vector in Torch because it is still a two-dimensional Tensor.

Create a column vector:

Matlab	`v = [9; 7; 6; 8]`
Torch	`v = torch.Tensor({{9}, {7}, {6}, {8}})`

But, this is not used as a vector in Torch because it is still a two-dimensional Tensor.

Create a one-dimensional tensor:

Matlab	*Not available*
Torch	`t = torch.Tensor({9, 7, 6, 8})`

Matlab does not have *one-dimensional tensors*; this can be verified by running `ndims([9, 7, 6, 8])`. On the other hand, for many operations that Matlab uses row or column vectors, Torch uses a one-dimensional tensor.

Access an element in a vector or one-dimensional tensor:

Matlab	`v(2)`
Torch	`t[2]`

Second element of the vector is accessed.

[*] Matlab® is a registered trademark of The MathWorks, Inc.

Access an element from the end of a vector or one-dimensional tensor:

Matlab	`v(end-1)`
Torch	`t[-2]`

Second element from the end of the vector is accessed.

Access a range of elements in a vector or one-dimensional tensor:

Matlab	`v(2:4)`
Torch	`t[{2,4}]`

Second to fourth elements of the vector are accessed.

Access an element in a Matrix:

Matlab	`m(2,3)`
Torch	`m[{2,3}]` or `m[2][3]`

In both cases, second row, third column element is accessed.

Access a row in a Matrix as a two-dimensional tensor:

Matlab	`m(2,:)`
Torch	`m[{2},{}]`

The returned row is a two-dimensional tensor.

Access a row in a Matrix as a one-dimensional tensor:

Matlab	*Not available*
Torch	`m[{2,{}}]` or `m[2]`

The returned row is a one-dimensional tensor.

Access a column in a Matrix as a two-dimensional tensor:

Matlab	`m(:,2)`
Torch	`m[{{},{2}}]`

The returned column is a two-dimensional tensor.

Access a column in a Matrix as a one-dimensional tensor:

Matlab	*Not available*
Torch	`m[{{},2}]`

The returned column is a one-dimensional tensor.

Access a range of elements in a Matrix:

Matlab	`m(2,2:4)`
Torch	`m[{{2},{2,4}}] or m[{2,{2,4}}]`

The second to fourth columns of the second row are returned. In Torch, there is a slight difference between using `index` (e.g. `2`) or `{index}` (e.g. `{2}`) for pointing to a singleton dimension. For `{index}`, the dimension of the returned tensor is same as the original tensor (e.g. tensor `m`). For `index`, the singleton dimension is removed, and the dimension of the returned tensor is one less than the original tensor (e.g. tensor `m`). Also, `{}` refers to all elements in that dimension. Finally, `-index` means `index`-th element from the end.

Forming Basic Tensors

Create a tensor over a range of values with step 1:

Matlab	`3:8`
Torch	`torch.range(3,8)`

Torch's result is a one-dimensional tensor.

Create a tensor over a range of values with an arbitrary step:

Matlab	`3:-1.9:-4.2`
Torch	`torch.range(3,-4.2,-1.9)`

Torch's result is a one-dimensional tensor.

Short description here:

Matlab	`Matlab code here`
Torch	`Torch code here`

Description of the details here.