# Theano:

Pros:

* Python library – language we all know and is easy to use and pickup.
* Good computational graphs
* Use of numpy allows handling of multidimensional arrays
* Native windows support
* Good for making algorithms from scratch
* Strong optimisation engine
* Many extra packages can be added through python
* One of the most extensible frameworks
* Very fast for small networks
* Large amount of documentations and tutorials

Cons:

* Difficult to install all important and recommended libraries (My opinion). Would not be easy for normal users. (This depends on the result for release)
* Possible bugs with AWS
* Long compile time for large models
* Low level API
* No multi GPU support

# Caffe:

Possibly caffe2 ?

Pros:

* Written in C++ (language we all know)
* Interface via Python and or Matlab
* C++ code written in highly optimised form to favour speed and space conservation
* Simpler for new users

Cons:

* Windows use currently not supported. Experimental version updated by community user.
* Simpler functionality than theano
* Not very flexible
* ConvNets usage drops usability significantly, not particularly within Caffe’s confort zone.
* Need to program in C++ and CUDA for large scale changes (CUDA unknown to us)
* Poor documentation
* Requires a lot of dependencies

# MatConvNet:

Pros:

* Aimed to be flexible yet simple.
* CUDA support built in, for efficiency with CNN’s

Cons:

* Is a matlab toolbox, matlab requires licensing

(Not as popular, hard to find info)

<https://deeplearning4j.org/compare-dl4j-torch7-pylearn>

<https://www.quora.com/Which-is-the-best-deep-learning-framework-Theano-Torch7-or-Caffe>

<https://arxiv.org/pdf/1511.06435.pdf>

<https://www.microway.com/hpc-tech-tips/deep-learning-frameworks-survey-tensorflow-torch-theano-caffe-neon-ibm-machine-learning-stack/>

<https://medium.com/@ricardo.guerrero/deep-learning-frameworks-a-review-before-finishing-2016-5b3ab4010b06>