TensorFlow

&

ModelSelection

Justin Chen
Boston University Machine Intelligence Community

TensorFlow Demo

https://github.com/martin-gorner/tensorflow-mnist-tutorial

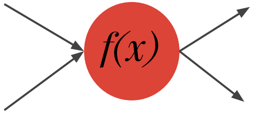
Machine Learning in a Nutshell

- Types of Learning:
  - Supervised - (example, label)
  - Unsupervised - (example)
  - Semi-supervised - (start supervised and continue unsupervised)
  - Reinforcement - (trial and error)
- Dataset: Training, Validation, Testing
  - Split (Pareto principle):
    - ~ 80% training (~70% training, ~10% validation)
    - ~ 20% testing
    - Percentages depend on how much data you have
- Tasks:
  - Classification
    - e.g. Given symptoms, classify illness
  - Regression
    - e.g. Given an image of the road, how many degrees to steer car

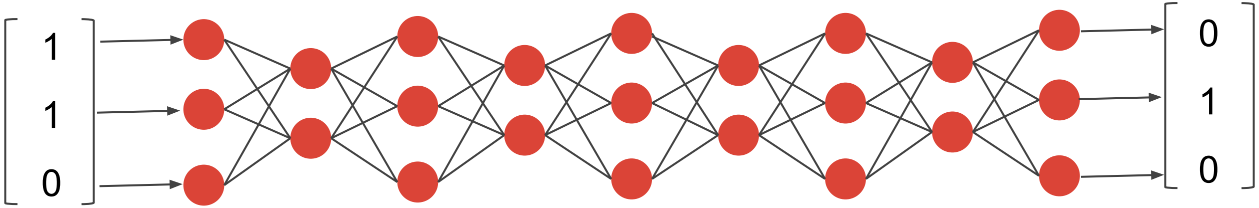
Yann LeCun’s Deep Learning black forest cake

Cherry	Reinforcement Learning
Frosting	Supervised Learning
Inside	Unsupervised Learning

Deep Learning in a Nutshell



- Artificial Neuron:
  - Combine neurons to get a network
    - Represent as a **computation graph** - artificial neural network (ANN)
- Basic Components of an ANN:
  - Input layer, Hidden layer(s), Output layer
  - Represented as matrices
  - More than 2 hidden layers considered “**deep**”, hence deep neural network (DNN)
- Good with high dimensional and nonlinear data - not a silver bullet
- Optimize weights according to cost/objective function SGD and BP



Deobfuscating Deep Learning

Frameworks, Abstractions, Open Source

Deep Learning Frameworks

torch

Microsoft
CNTK

dmlc
mxnet

TensorFlow

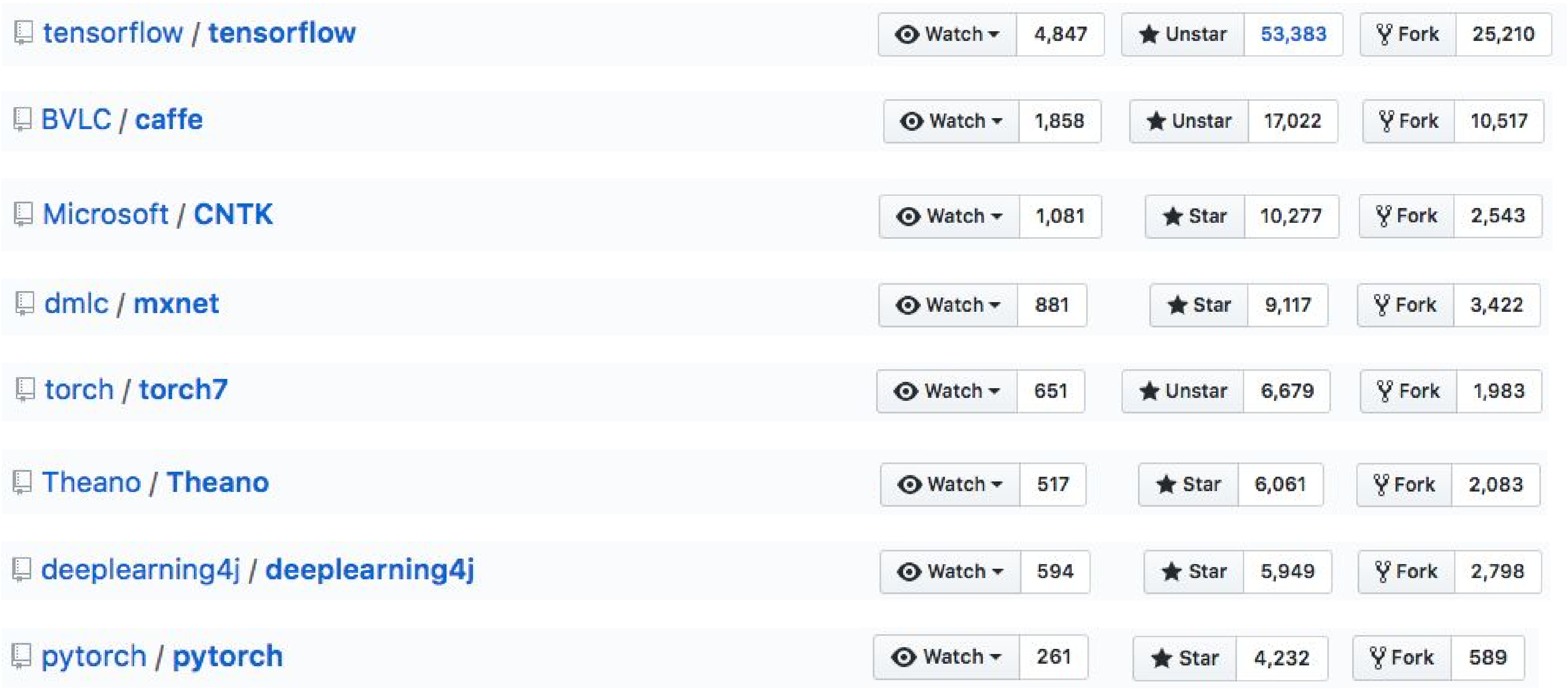
PYTORCH

theano

DL4J
DEEPLEARNING4J

Caffe

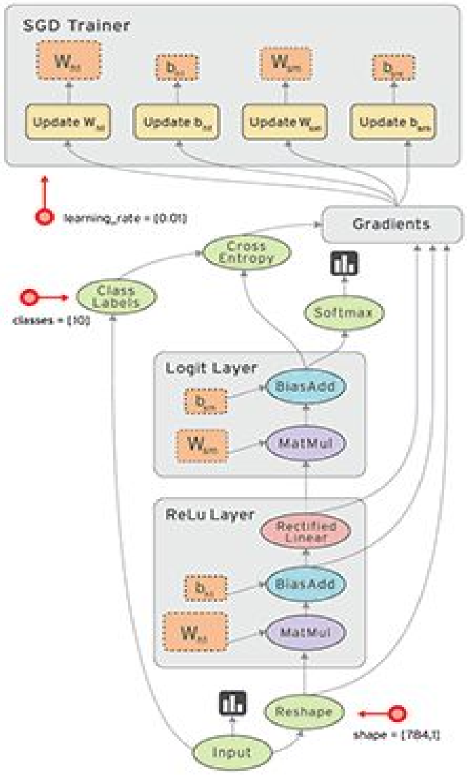
Why TensorFlow?



tensorflow / **tensorflow**	Watch ▾	4,847	★ Unstar	53,383	Fork	25,210
BVLC / **caffe**	Watch ▾	1,858	★ Unstar	17,022	Fork	10,517
Microsoft / **CNTK**	Watch ▾	1,081	★ Star	10,277	Fork	2,543
dmlc / **mxnet**	Watch ▾	881	★ Star	9,117	Fork	3,422
torch / **torch7**	Watch ▾	651	★ Unstar	6,679	Fork	1,983
Theano / **Theano**	Watch ▾	517	★ Star	6,061	Fork	2,083
deeplearning4j / **deeplearning4j**	Watch ▾	594	★ Star	5,949	Fork	2,798
pytorch / **pytorch**	Watch ▾	261	★ Star	4,232	Fork	589

Building and Training Computation Graphs

1. Hyperparameters
2. Define topology
3. Define cost
4. Optimizer
5. Initialize and train
6. Test accuracy

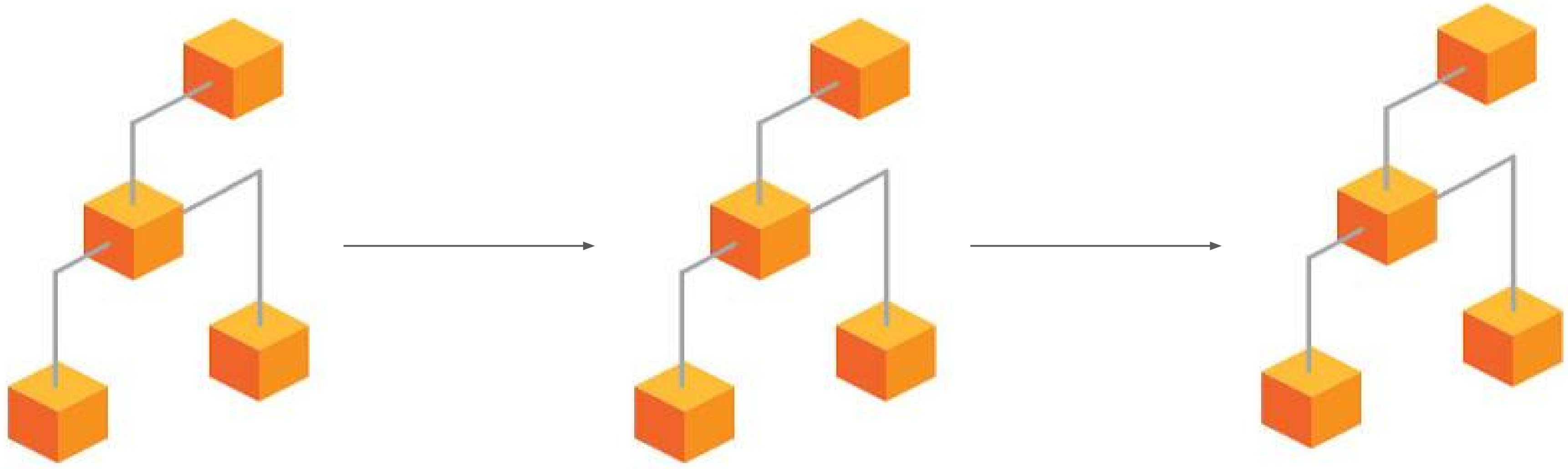


https://www.tensorflow.org/images/tensors_flowing.gif

Fold

Arbitrary Size Data, Dynamic Computation Graphs, More Abstractions

TensorFlow Fold Combinator Library



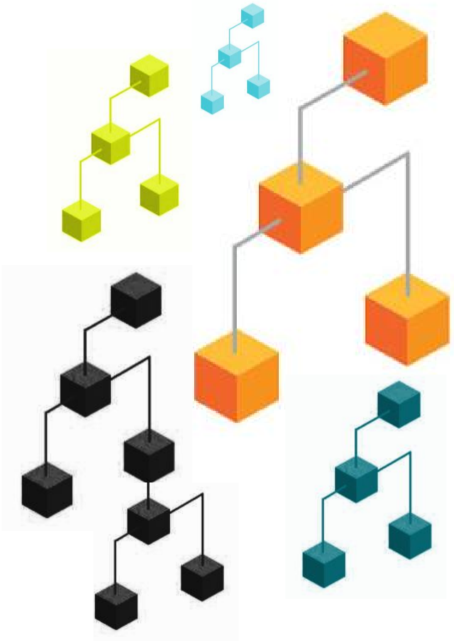
Mini-batch of Python objects

Fold

TensorFlow Tensors

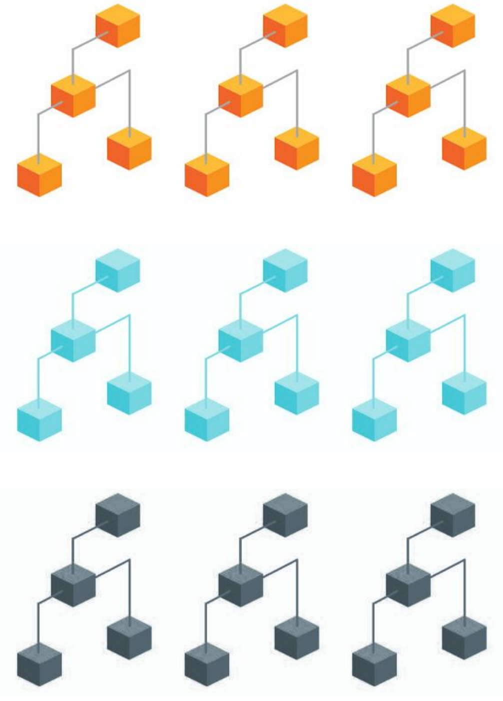
Fold: Dynamic Computation Graph

- TensorFlow builds **static** computation graphs
  - Input must be **padded** or **truncated** to the same dimensions
- Fold builds a **different** computation graph for each input to accommodate different sizes
- **Directed acyclic computation graphs**
- Can handle **tree-structured** and **graph-structured** data



Fold: Dynamic Batching Algorithm

- Assumes different computation graph for every input
- Takes set of computation graphs as input
- Batch same operations that occur at same depth
- Schedules operations automatically



Fold Blocks

- Basic unit is a `td.Block`
  - I/O: tensors, tuples, lists, dictionaries, combinations
- Block arranged into a **tree**
  - **Primitive types** form leaves of tree: `td.Scalar()` and `td.Vector((shape))`
- Block **composition** for more complex computation
  - e.g. `td.Vector(784) >> td.Function(td.FC(100))`
- `td.Fold()` and `td.Reduce()` apply operation to given block
- After defining and composing blocks, pass to `td.Compiler()`
  - Outputs tensors for use in TensorFlow
  - Pass tensors to optimizer and use as normal TF computation graph

Space of Possible Architectures

Exploration v. Exploitation

Neural Network Architectures

Presentation tomorrow at MIT’s Machine Intelligence Community

Topic: Revisiting Neuroevolution of Augmenting Topologies

Location: 56-154 (building 56, room 154)

Time: 5 PM EST

Incentive: Free food, machine learning, evolutionary algorithms

References

[0] Abadi, Martín, et al. "TensorFlow: A system for large-scale machine learning." *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI). Savannah, Georgia, USA*. 2016.

[1] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems*. 2012.

[2] Dumoulin, Vincent, and Francesco Visin. "A guide to convolution arithmetic for deep learning." *arXiv preprint arXiv:1603.07285* (2016).

[3] Looks, Moshe, et al. "Deep learning with dynamic computation graphs." *arXiv preprint arXiv:1702.02181* (2017).