Interdisciplinary
Center for Applied
Machine Learning

Applied
Machine Learning
Academy

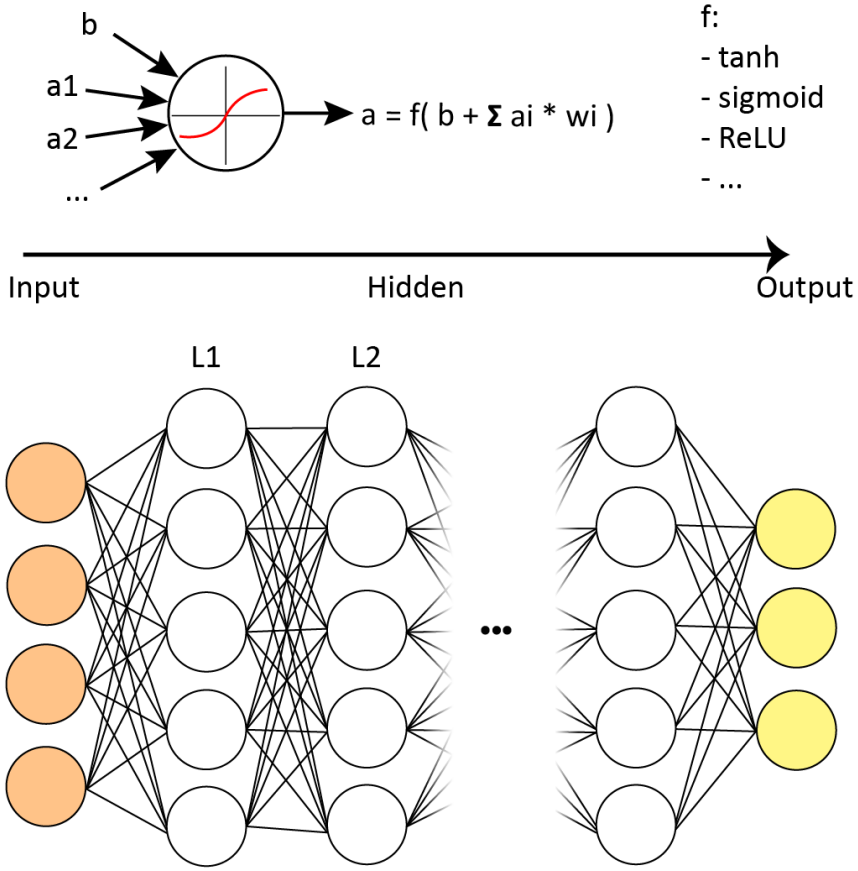
Programming Languages and Frameworks for Data Science

AMA / ICAML - 01.10.2019

Deep Learning

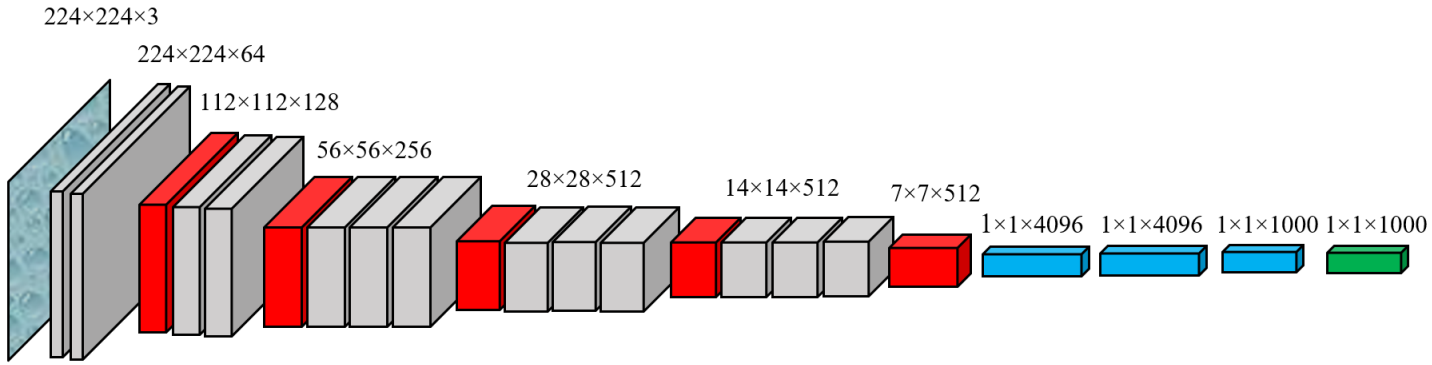
Deep Learning

- Subfield of machine learning
  - Commonly (wrongly) known as *Artificial Intelligence*
- Based on Deep Neural Networks
  - Artificial neurons, usually in layers
  - Very complex mappings can be modeled
- Training requires:
  - Specific hardware (GPU / TPU)
  - **Many training examples** (labeled data)
  - Time (and experience)
- Training based on gradient descend
  - Minimization of a differentiable loss function
- Specific architectures / loss functions for different data-types and applications



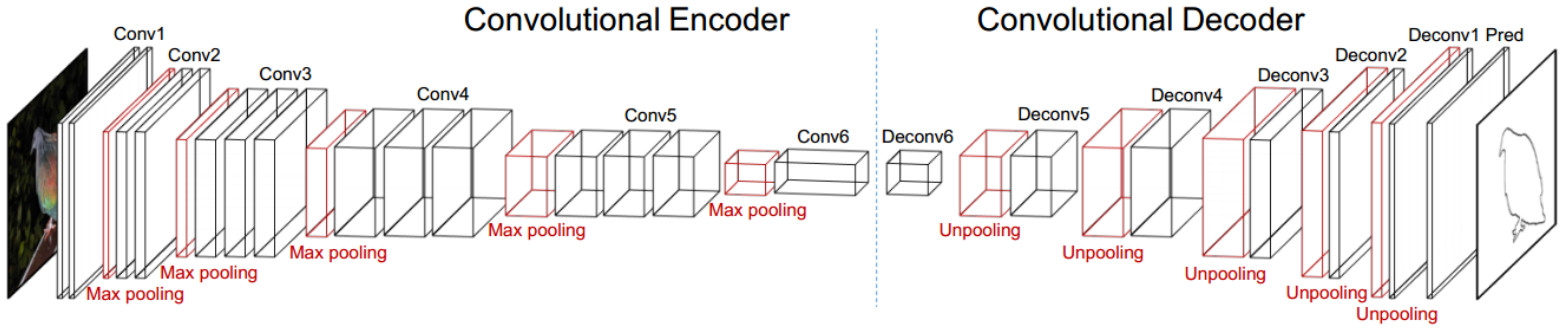
DL –Applications / Architectures

- **Image classification**
  - Convolutional Neural Network
  - Convolutions and
  - Pooling / downsampling
  - Predicts one class per image



[source: wikipedia.org/wiki/File:VGG_neural_network.png]

- **Image segmentation**
  - Fully Convolutional Network
  - Encoder similar to CNN
  - Decoder uses upsampling via unpooling or transposed convolution
  - Predicts one class per pixel

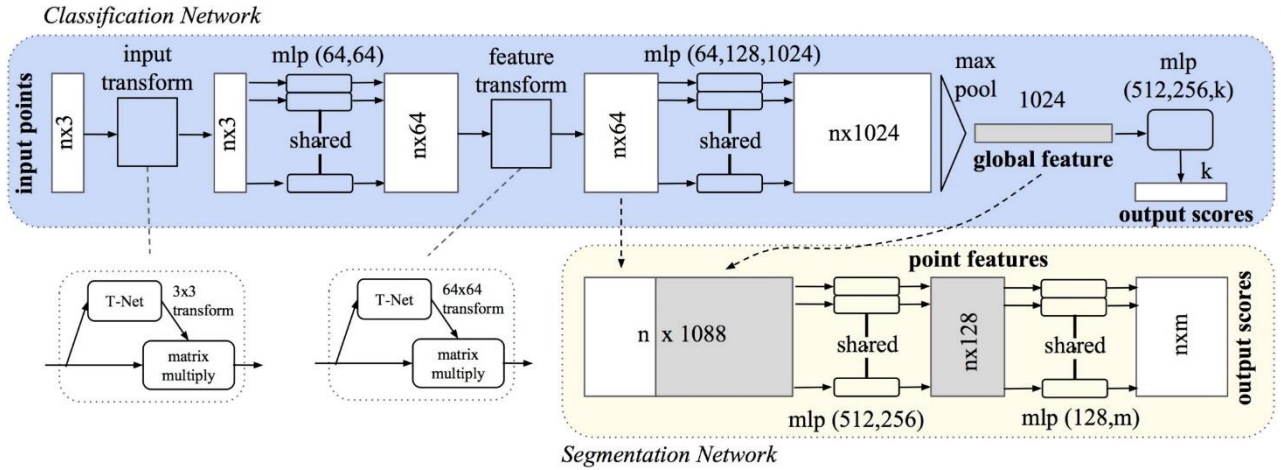


[source: nsarafianos.github.io/icip16]

- Both architectures can also be used for regression

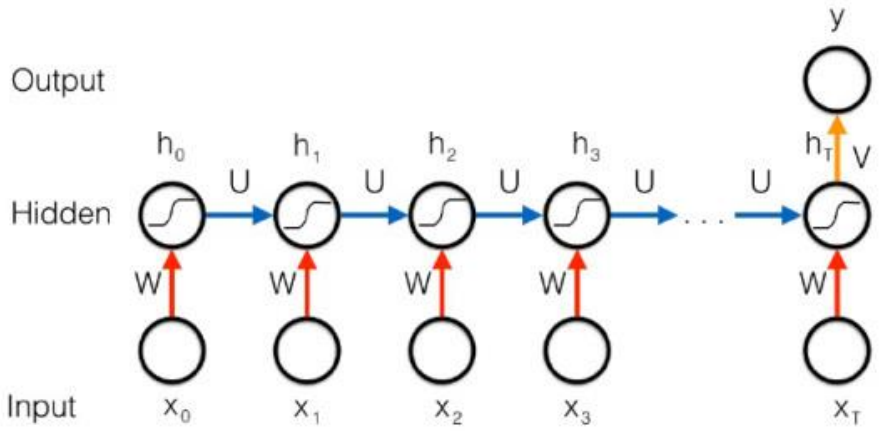
DL – Applications / Architectures

- **Point cloud classification**
  - E.g. PointNet
  - Requires special layers for unordered data
  - Predicts one class for cloud/ one class per point



[source: http://stanford.edu/~rqi/pointnet/images/pointnet.jpg]

- **Natural language processing**
  - E.g. for language understanding or translation
  - Due to variable input length: Recurrent Neural Networks
  - Process ‚word-by-word' with memory



[source: naviglinlp.blogspot.com/2018/04/lecture-10-06042018-recurrent-neural.html]

….. and there are MUCH more!

Deep Learning – Use cases

Researcher

- Training from scratch (or retraining)
- Optimizes network for own needs
- Needs access to all details
- Knows/sets all parameters
- Is interested in fast training

→ Requires low-level API

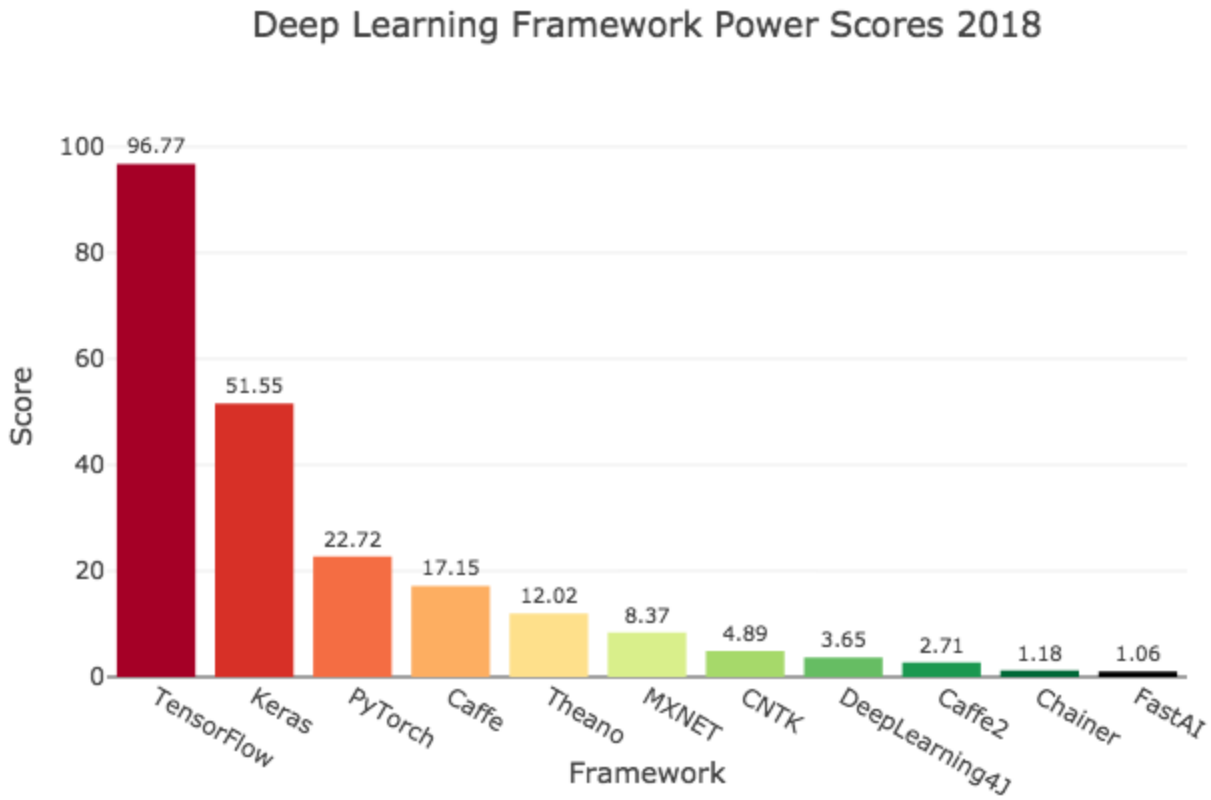
Common user

- Mostly retrains pre-trained models based on well studied architectures
- Usually doesn‘t modify the code
- Parameters and details are less important
- Is interested in fast inference

→ Requires high-level API + optimized deployment

Frameworks for Deep Learning

- Many frameworks - few popular
- Mostly provide similar functionality
  - All can be used for ‚usual‘ cases
  - Choice depends on restrictions / preference
- Most essential features:
  - GPU / TPU support
  - Automated gradient calculation
  - Frequently used functions / layers
- Most supported language: **Python**
- Powered by *big players*:
  - TensorFlow (Google Brain)
  - PyTorch / Caffe2 (Facebook)
  - Cognitive Toolkit CNTK (Microsoft)



towardsdatascience.com/deep-learning-framework-power-scores-2018-23607ddf297a

DL – Frameworks and Languages

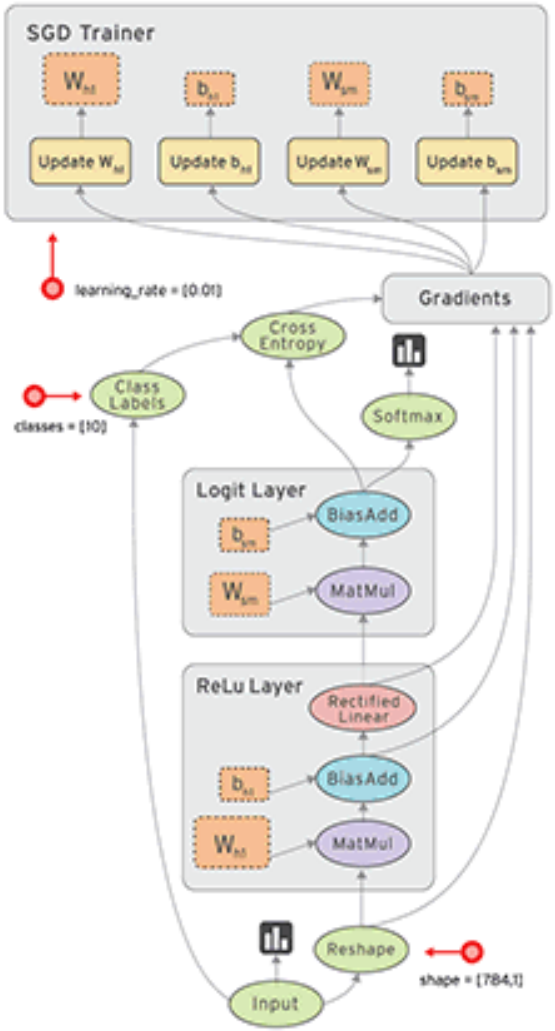
	Python	C++	R	Java	Matlab	Julia
TensorFlow	✓	✓	✓	(✓)		(✓)
Keras	✓		(✓)			(✓)
PyTorch	✓					(✓)
Caffe	✓	✓	(✓)		✓	(✓)
CNTK	✓	✓	(✓)			(✓)
MXNet	✓	✓	(✓)			✓
DeepLearning4J	(✓)		(✓)	✓		(✓)
Chainer	✓					
Caffe2	✓	✓				

(✓) : access via wrapper or Keras

- Most supported language: **Python**

TensorFlow

- Currently most popular Framework
  - Low-level
  - High-level (Keras)
- Based on two-step approach
  1. Define computation graph (including rules for updats etc..)
  2. Run graph on any hardware (either for training or prediction)
- Requires some time to get started
  - Concepts are not usual in ‚regular' programming
  - Huge amount of functions which are updated quite often
- Documentation and online support is very good
- Additionally offers:
  - Easy to use architectures and trained models
  - Complex, but powerful visualization tool: TensorBoard



tensorflow.org

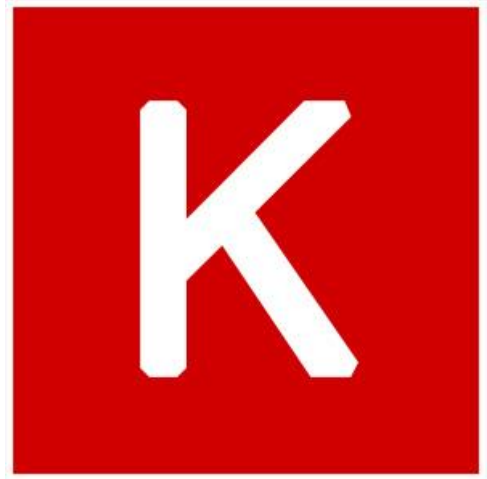
PyTorch

- Growing attention
- Low- and high-level functions
- Uses dynamic graphs:
  - Allows to change computation graph between iterations
  - Makes usage simpler and more like ‚usual' programming
- Worse documentation and less functionalities than TF
- Easy access to pretrained models

pytorch.org

Keras

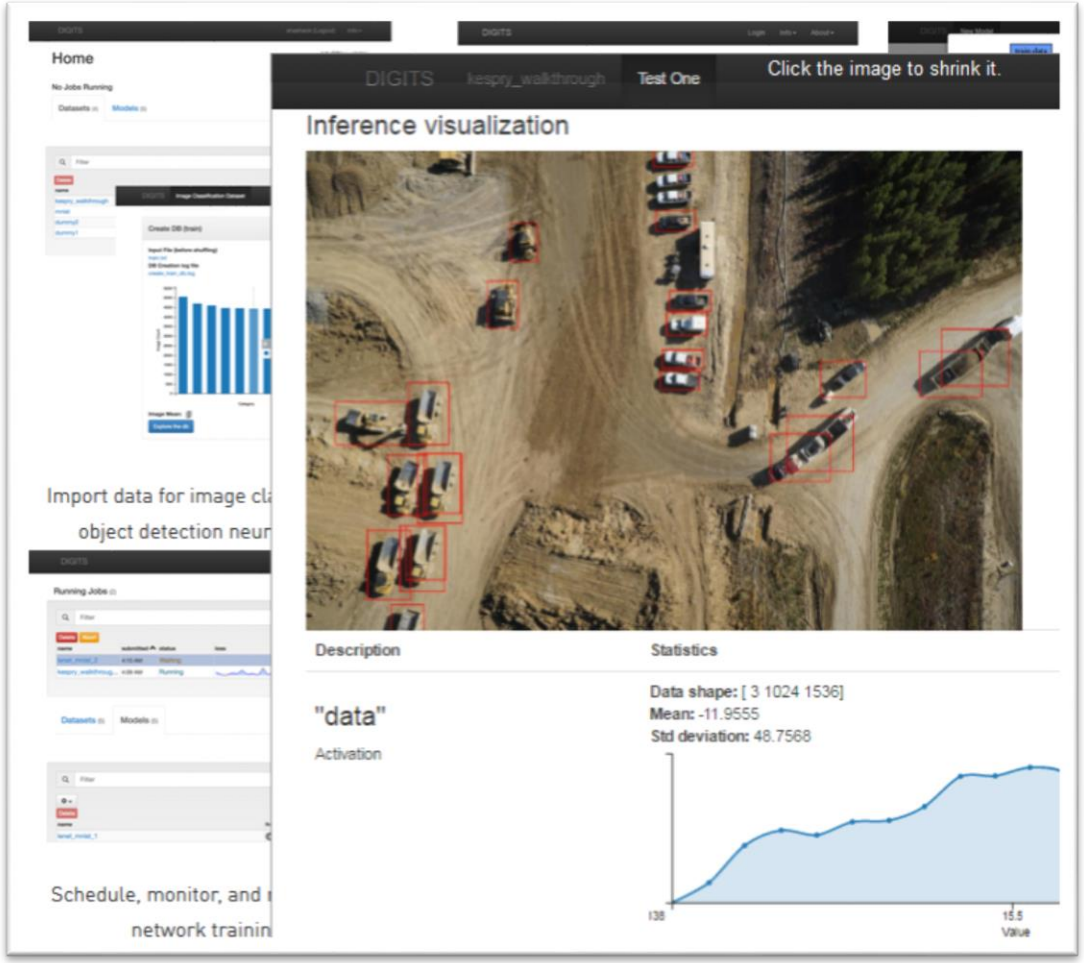
- High-level API for:
  - Tensorflow
  - Theano
  - CNTK
  - PlaidML
- Allows training models in very few lines of code
- Currently **not** so well documented
- Good for beginners, that just want to replicate models
- Often hides relevant information from user
- Will be default API in TensorfFlow 2.0



keras.io

Digits

- Deep Learning GPU Training System (Nvidia)
- Server application / access via browser
- Features:
  - Model choice (including simple modifications)
  - Data overview and processing
  - Training / Evaluation / Prediction
  - Visualization of Graph / Results
- Very high-level!
- Does not require any programming skills



developer.nvidia.com/digits