\Orchestrating a brighter world NEC

TORCH: A Powerful Machine Learning Development System

March 17, 2016

Hans Peter Graf

NEC Laboratories America
Relentless passion for innovation

www.nec-labs.com

Overview

Introduction of Torch

Deep Learning:

- Introduction
- Image Analysis
- Text Analysis
- Examples of applications

Implementations:

- Benchmarks

Conclusions

torch

http://torch.ch/

A SCIENTIFIC COMPUTING FRAMEWORK FOR LUAJIT

Easy to use thanks to a fast scripting language: Lua (C-like syntax)
Efficient: C, CUDA implementations highly optimized for performance.

A summary of core features:

- Extensive support for Deep Learning and other machine learning techniques
- Develop sophisticated models with a few lines of script code
- Very efficient interface to C, via LuaJIT
- Linear algebra routines
- Numeric optimization routines
- Lots of routines for indexing, slicing, transposing, ...
- Efficient support for GPU, OpenMP, AVX, …
- Embeddable, with ports to iOS, Android and FPGA backends

Large ecosystem with powerful packages in machine learning, vision, text analysis, signal processing, parallel processing, image, video, audio

Torch: Background

History:
2000: Started at IDIAP (Switzerland), Torch 1, 3, 5
2006: Continued at NEC Laboratories America, Torch 5, 7
2011, 2012: Adopted also by New York University, DeepMind

Check out: **https://github.com/torch/torch7/blob/master/COPYRIGHT.txt**

Copyright (c) 2011-2014 Idiap Research Institute (Ronan Collobert)
Copyright (c) 2012-2014 Deepmind Technologies (Koray Kavukcuoglu)
Copyright (c) 2011-2012 NEC Laboratories America (Koray Kavukcuoglu)
Copyright (c) 2011-2013 NYU (Clement Farabet)
Copyright (c) 2006-2010 NEC Laboratories America (Ronan Collobert, Leon Bottou, Iain Melvin, Jason Weston)
Copyright (c) 2006 Idiap Research Institute (Samy Bengio)
Copyright (c) 2001-2004 Idiap Research Institute (Ronan Collobert, Samy Bengio, Johnny Mariethoz)

Open distribution: MIT license, no strings attached (check contributed components!)

Torch 7 Today:
Very popular Open Source system for Machine Learning research and commercial developments.
- Industry: NEC, Facebook, Twitter, Google DeepMind, many others
- Large number of universities
- Popular among serious developers

Software Architecture

Script: Lua

Lua function binding (FFI)

C: Deep Learning, SVM, Statistics, …

Low-level libraries: CuDNN, MKL, …

BLAS, LAPACK, CUDA, OpenMP, …

Hardware: CPU, GPU, FPGA

Multi-core, Many-core, Cluster, …

Develop in script: fast, easy

User does not see this

Packages

Software is organized in packages for a wide range of different applications
https://github.com/torch/torch7/wiki/Cheatsheet

Core Math	Visualization	Utility libraries
Data formats I/O	Sensor I/O	Databases
Machine Learning	Computer Vision	NLP
Parallel Processing	CUDA	OpenCL
Images	Videos	Audio
Asynchronous	Networking	Security
Alternative REPLs	Interfaces to third-party libs	Reinforcement Learning
Miscellaneous		

Many more packages are available via Lua's package manger: luarocks
Check out what's available here: https://github.com/torch/rocks

Torch use in industry: For example Facebook

Y. LeCun (head Facebook AI):
"Torch is for research in deep learning; Caffe is OK for using ConvNets as a "black box" (or a gray box), but not flexible enough for innovative research in deep learning. That's why Facebook and DeepMind both use Torch for almost everything."

Facebook releases Open Source:
https://github.com/facebook/fblualib

- **C++ LuaUtils** is a collection of C++ utilities useful for writing Lua extensions
- **fb.debugger** is a full-featured source-level Lua debugger.
- **fb.python** is a bridge between Lua and Python, allowing seamless integration between the two (enabling, for example, using SciPy with Lua tensors almost as efficiently as with native numpy arrays;
- **fb.thrift** is a library for fast serialization of arbitrary Lua objects using Thrift. Requires Torch.
- **fb.mattorch** is a library for reading and writing Matlab .mat files from Torch without having Matlab installed.
- **More:** see web site

Deep Learning: Recent Boom

Breakthrough in Big Data Analytics

- Deep Learning has become a dominant approach for many tasks
- Industry has adopted Deep Learning widely: **NEC, Google, Microsoft, Amazon, Facebook, Baidu, IBM Watson, ….**
- **Broad Open Source ecosystem is developing**
- **Many large-scale data sets become available**

Related: Strong international efforts in research on Brain modeling and **Neuromorphic Computing**

- **European Union:** Human Brain Project $1.3Billion, 10 years; develop a Neuromorphic Computing Platform to develop models of brain operation.
- **US:** BRAIN Initiative; 10 year initiative.

Note: Deep Learning is NOT a model of how the brain learns!!

Deep Learning: A Brief History

1940 | 1960 | 1980 | 1990 | 2000 | 2010

Hebbian Learning

Perceptron

Hubel/Wiesel

Backprop Algorithm

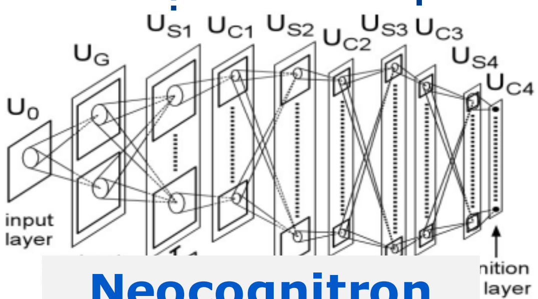
Neural Network Boom

SVM

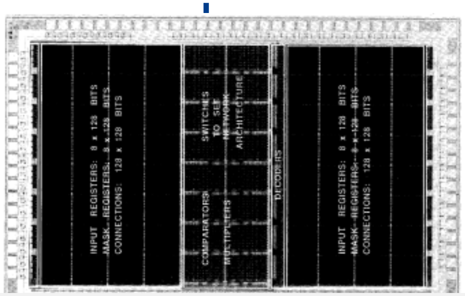
Support Vector Machine Boom

Deep Learning Boom

Several Layers	One Layer	Many Layers
Neuromorphic inspiration	Statistics	Empirics



Neocognitron (Fukushima)



Neural Net chips

Large scale learning, GPU

Deep Learning: Principle

Why are Deep Learning networks good?

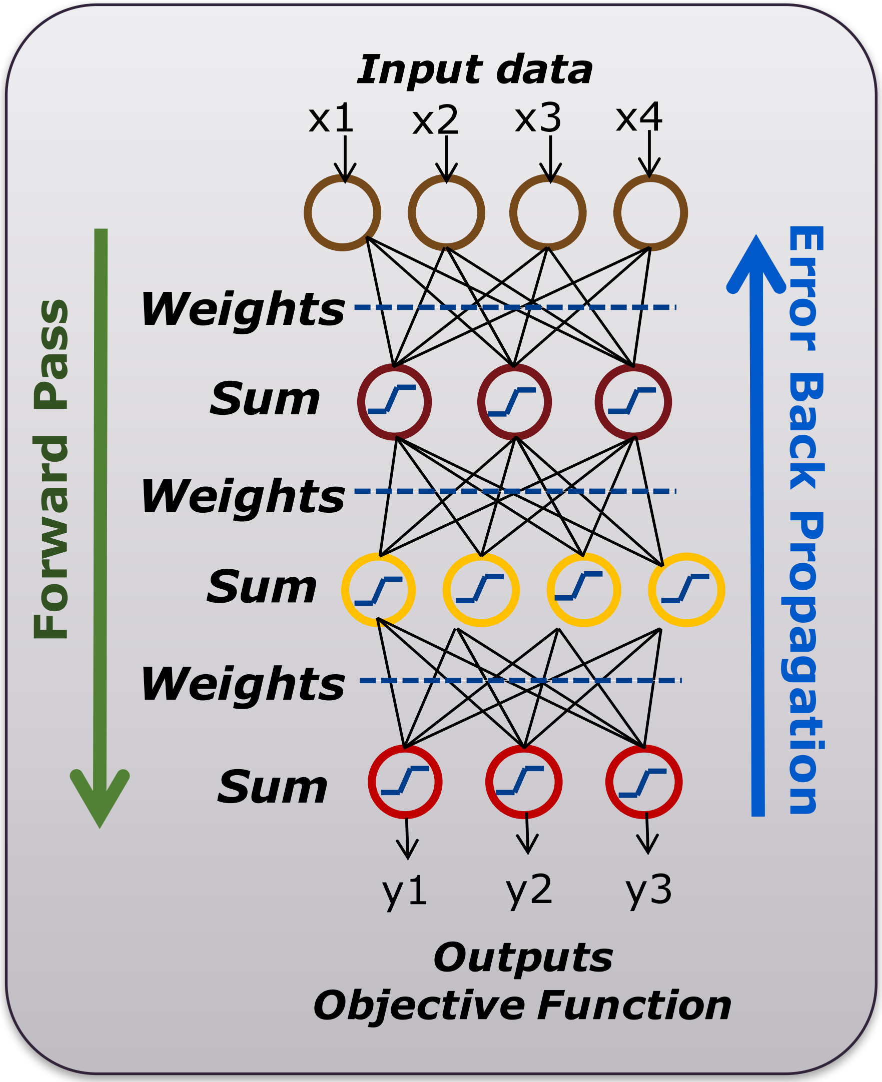
- **Top performance:** Speech vision, text analysis, robotics, …
- **Feature Learning:** Learn features automatically.
- **Efficiency:** Deep Learning networks can be more efficient than Support Vector Machines. ***Deep versus Wide***

Problems with Deep Learning networks?

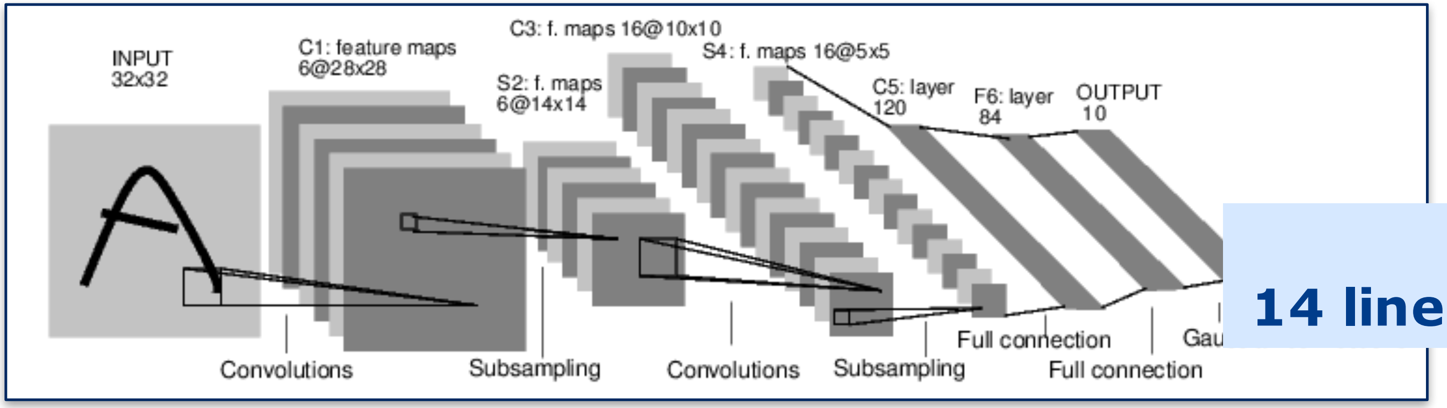
- Learned features cannot be interpreted
- Learning is very compute intensive (often training for weeks)
- Lots of heuristics to make them work; non-convex optimization.
- Poor theoretical foundation

No great new algorithms – but faster computers; can train with more data

Multi-Layer Neural Net



Build a Deep Learning Network in Torch 7



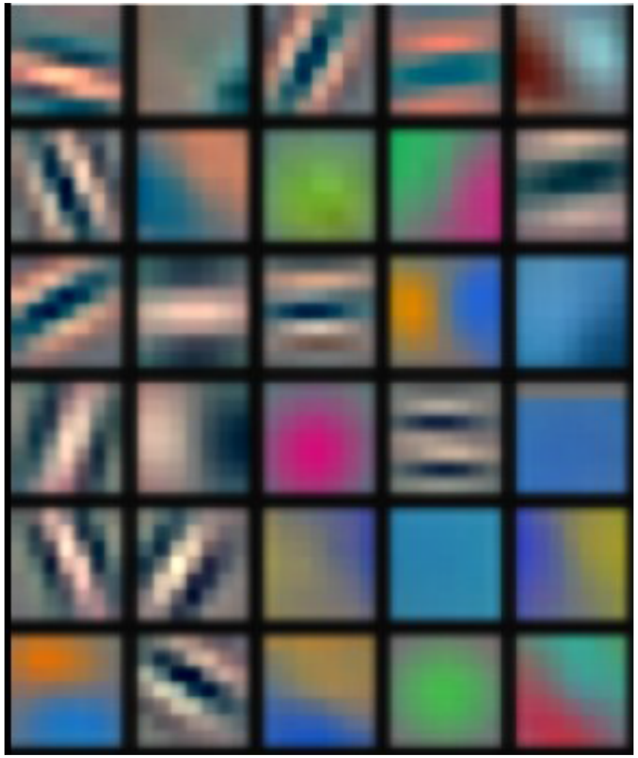
LeNet-5:
14 lines of script code

Lua command	Comments
net = nn.Sequential()	Define a network
net:add(nn.SpatialConvolution(1, 6, 5, 5))	1 input image, 6 output planes, 5x5 kernels
net:add(nn.ReLU())	Non-linearity: Rectified linear units
net:add(nn.SpatialMaxPooling(2,2,2,2))	Max-pooling over 2x2 windows; stride: 2 pixels
net:add(nn.SpatialConvolution(6, 16, 5, 5))	Convolution layer: 6 input, 16 output planes, 5x5 kernels
net:add(nn.ReLU())	Non-linearity: Rectified linear units
net:add(nn.SpatialMaxPooling(2,2,2,2))	Max pooling over 2x2 windows; stride: 2 pixels
net:add(nn.View(16*5*5))	Convert planes (3D tensor of 16x5x5) to 1D vector
net:add(nn.Linear(16*5*5, 120))	Fully connected layer: input 400, output 120
net:add(nn.ReLU())	Non-linearity: Rectified linear units
net:add(nn.Linear(120, 84))	Fully connected layer: input 120, output 84
net:add(nn.ReLU())	Non-linearity: Rectified linear units
net:add(nn.Linear(84, 10))	
net:add(nn.LogSoftMax())	Output: Log probability

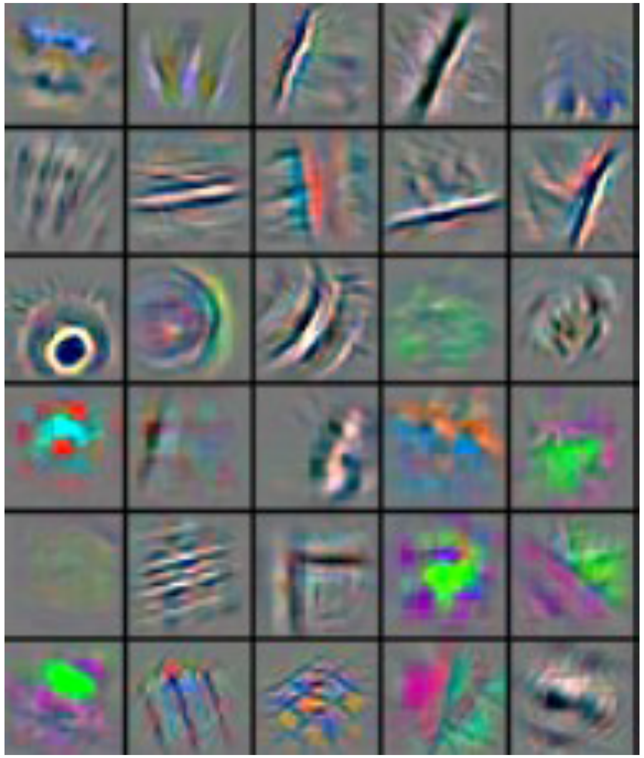
Advantage of Deep Learning: Feature Learning

- Standard approach in image recognition: Feature extraction with SIFT, HOG or similar features (pyramids).
- In CNN convolution kernels develop as feature detectors.
- From input towards output features tend to become more high level

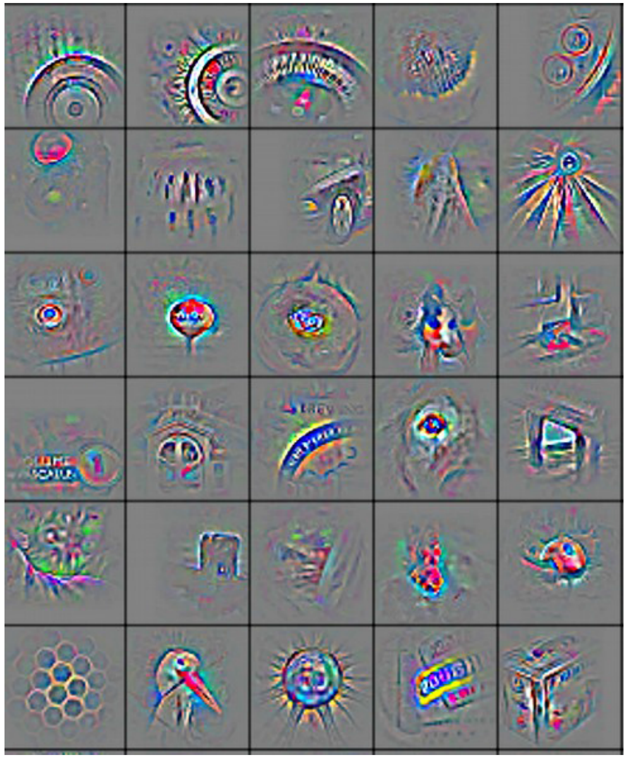
Low level features



Mid level features

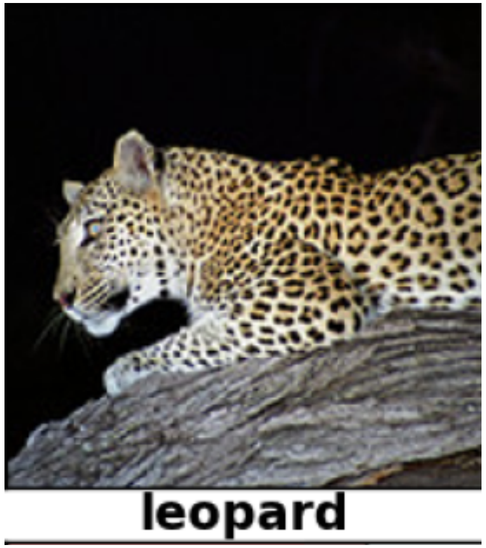


High level features

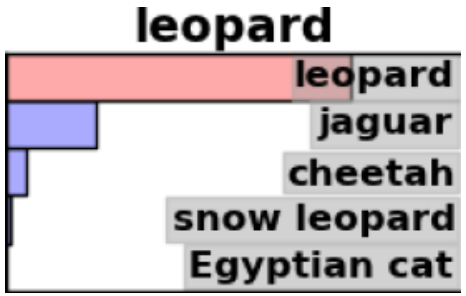


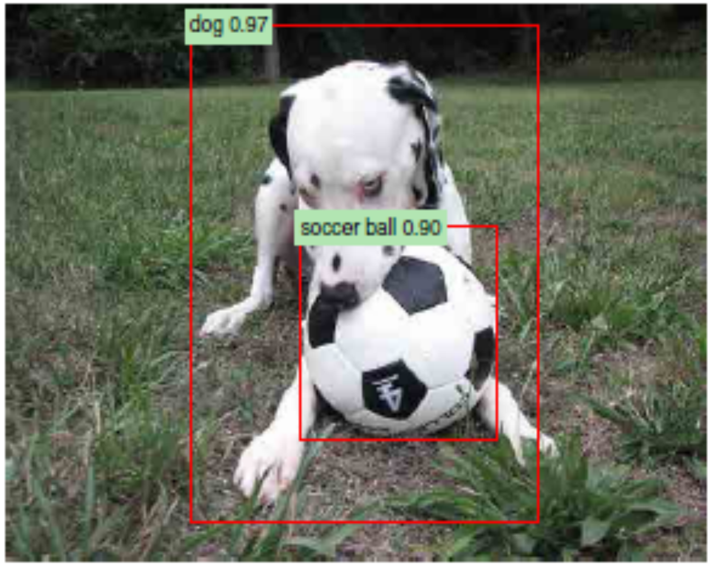
Visualizing and Understanding CNN: Zeiler & Fergus 2013

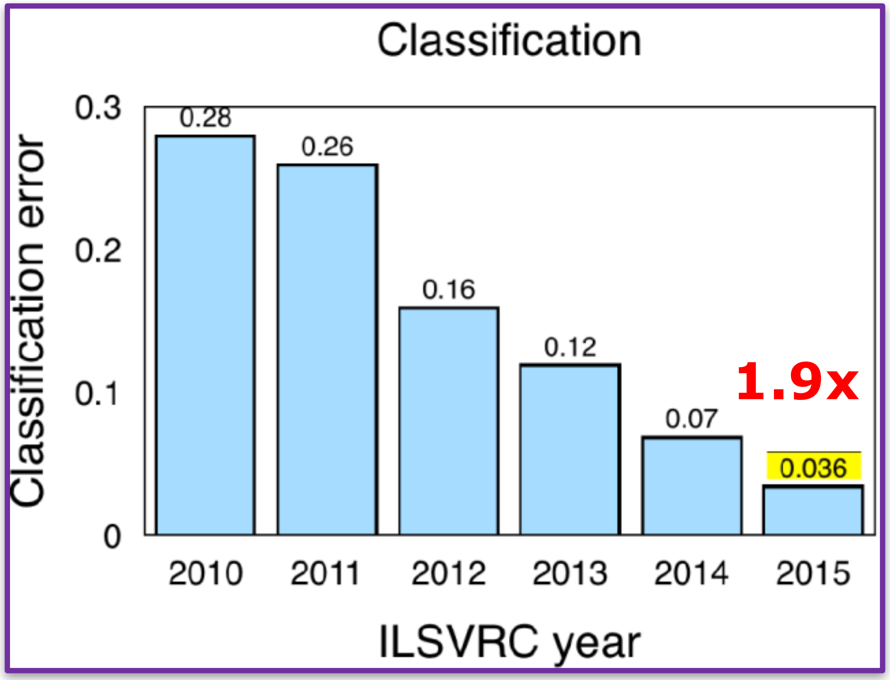
Image Analysis: Spectacular Progress

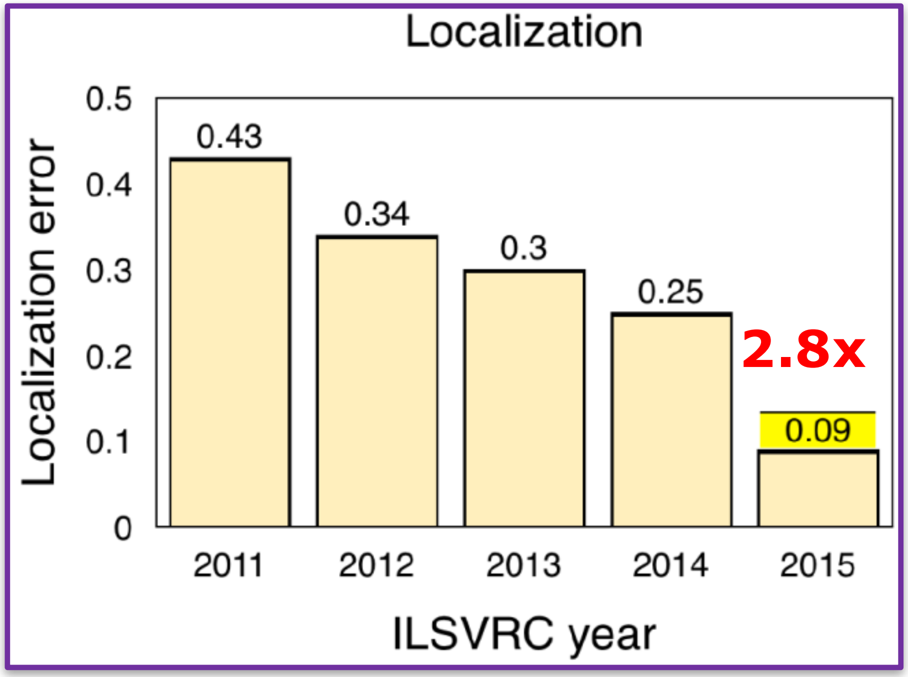


ILSVRC Competitions
1,000 classes

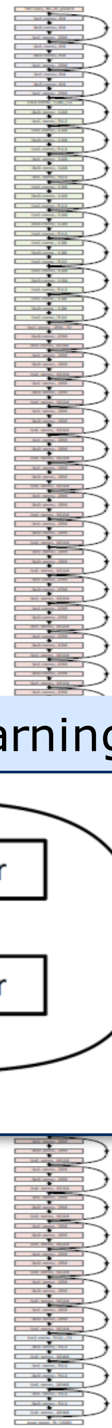








Revolution of Depth

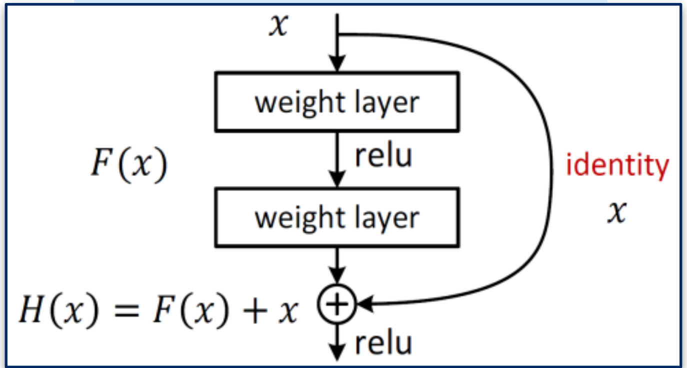


Where is drastic improvement in accuracy coming from??

- Biggest contribution comes from faster computers that allow training larger networks
- Considerable improvements are due to data augmentation (artificially generated training examples)

Recent trend: deeper and narrower networks
Such networks used to be difficult to train, since gradients of deep nets tend to disappear. But normalization of gradients at each layer and other operations resolve this problem

Residual Learning



ILSVRC Winners

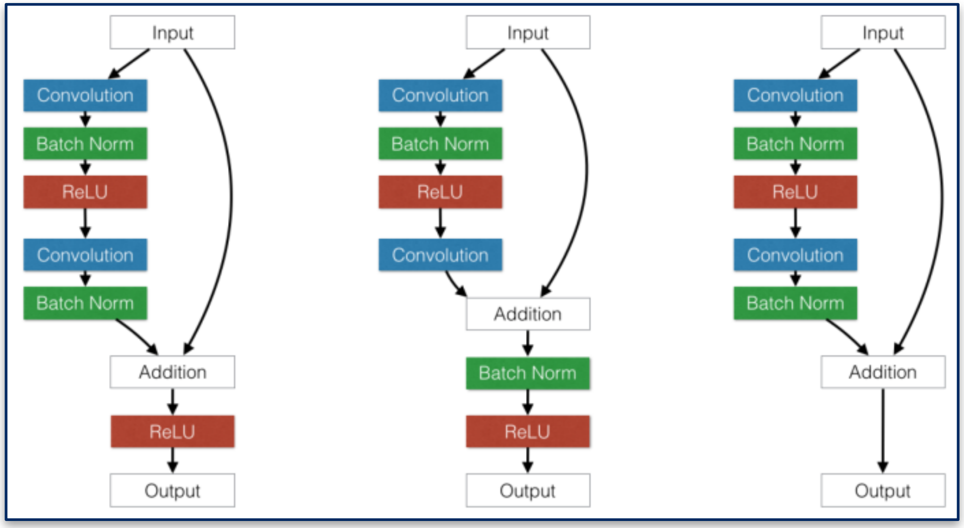
2012: AlexNet	2014: VGG	2015: ResNet
8 Layers	19 Layers	152 Layers

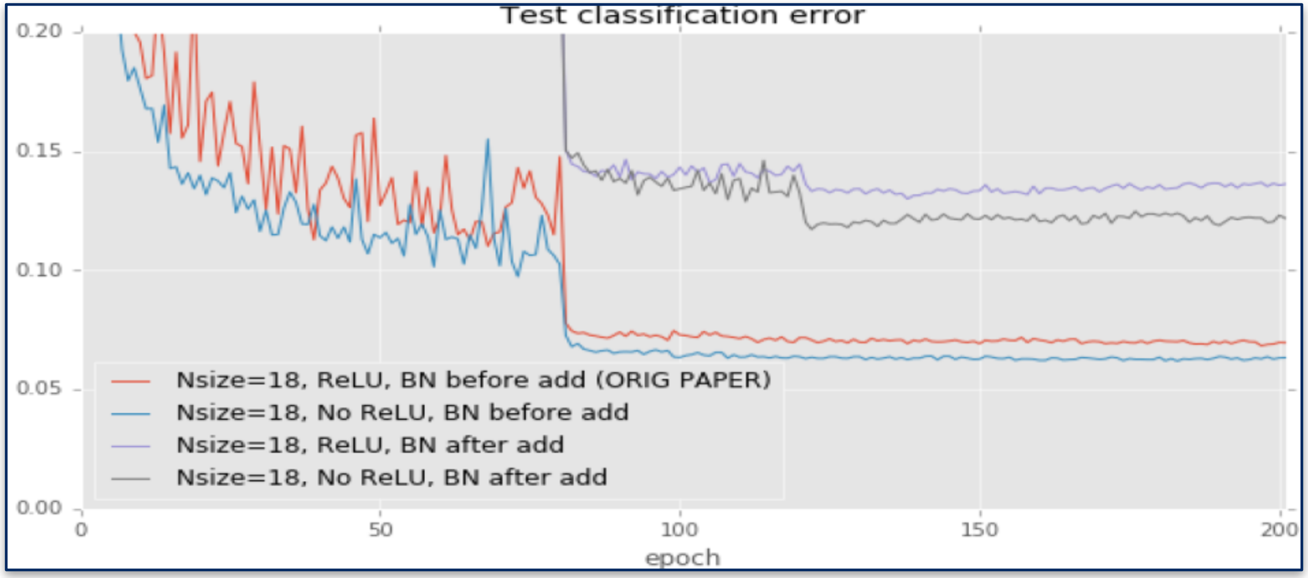
Improving Performance of Deep Learning

- **Deep Learning is improved empirically**
- **Requires lots of tests.**
- **Basically no theory to guide the development**
- **Lots of trial and error**

Impressive progress is made thanks to a large research community that shares results with open source releases and shared data sets.

Varying details of the network architecture





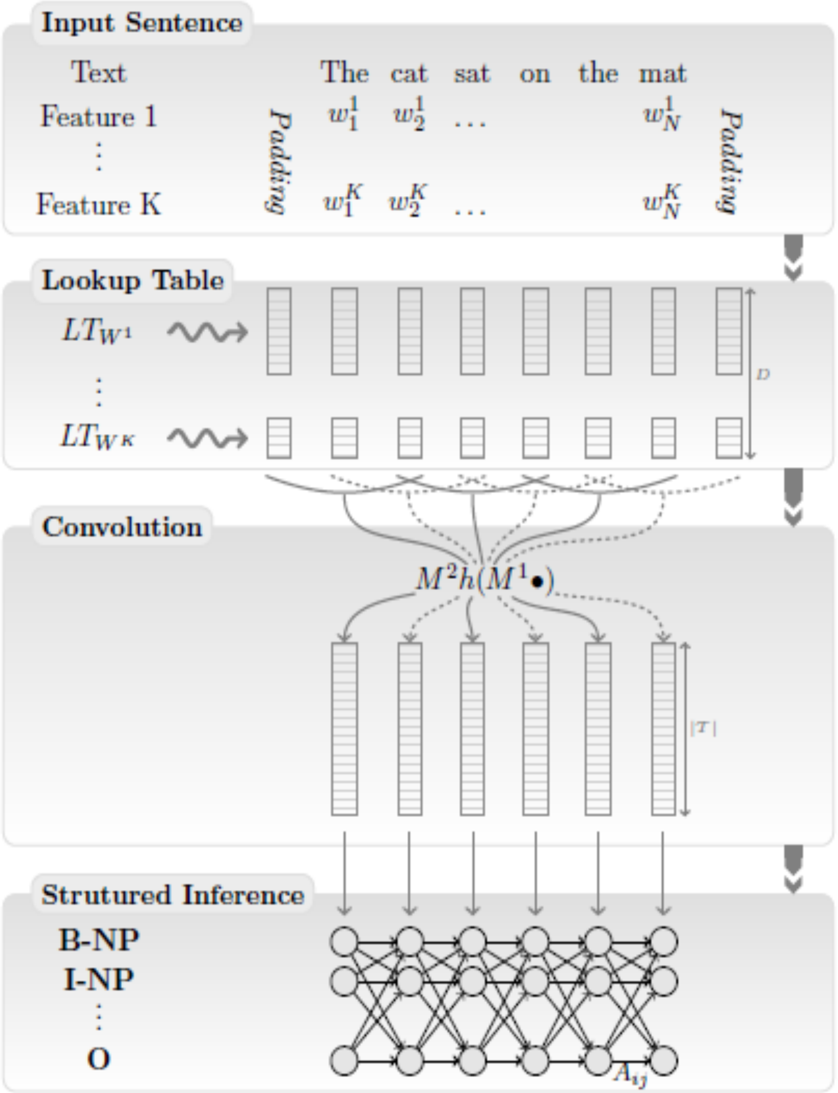
http://torch.ch/blog/2016/02/04/resnets.html

Torch for Natural Language Processing

Much of the data on the Web and everywhere are unstructured text
- This requires syntactic analysis first, followed by semantic interpretation
- Many functions are supported in Torch with ready-made libraries

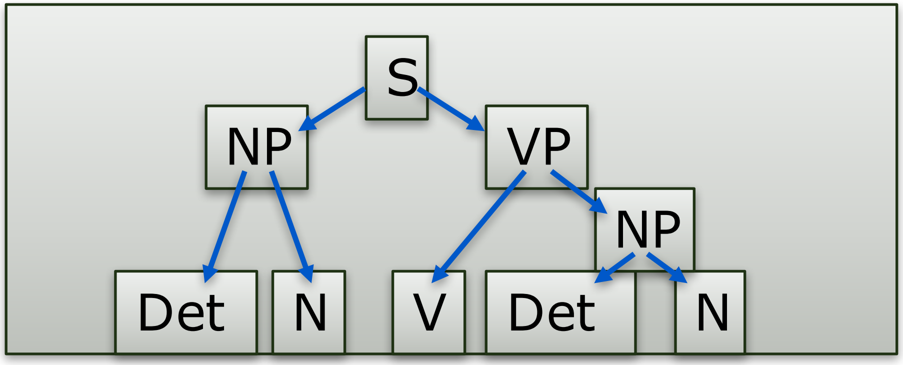
- **nn** - Neural language models can be implemented using the nn package.
- **rnn** - Recurrent Neural Network library with Recurrent and LSTM models that can be used for language modeling.
- **dp** - Includes Neural Network Language Model Tutorial and Recurrent Neural Network Language Model implementations using the Google Billion Words dataset.
- **senna** - Part-of-speech tagging, Chunking, Name Entity Recognition and Semantic Role Labeling, extremely fast
- **word2vec** - Ready to use word2vec embeddings in Torch. Provides example.

Deep Learning: Text Analysis



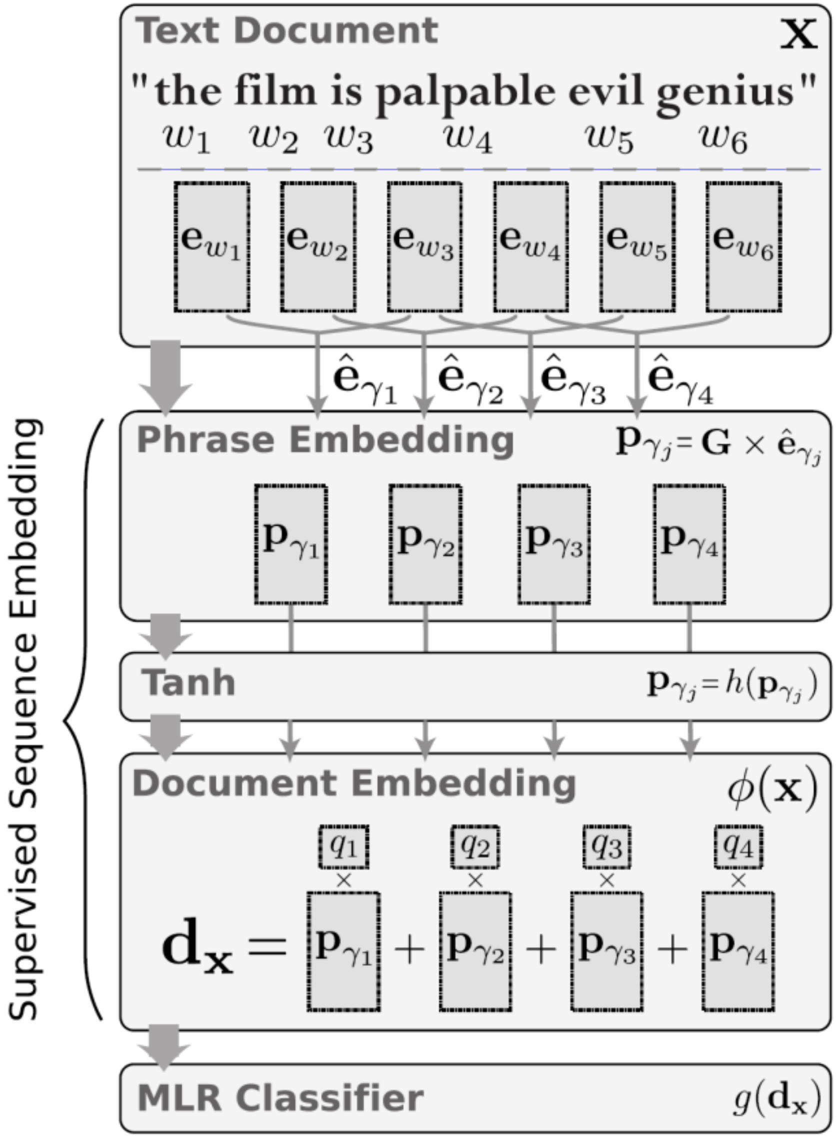
Syntactic Analysis

- Trained from raw text: No handcrafted features
- Tasks:
  - Part of Speech
  - Chunking
  - Name Entity Recognition
  - Semantic Role Labeling
  - Parsing
- Different languages: Just training with language corpus



Natural Language Processing (Almost) from Scratch, R. Collobert et al., JMLR 2011

Semantic Embedding of Sequences



Example: Sentiment Analysis
- Rank phrase how positive it is.
- Here: All phrases contain ‘good’ and ‘book’

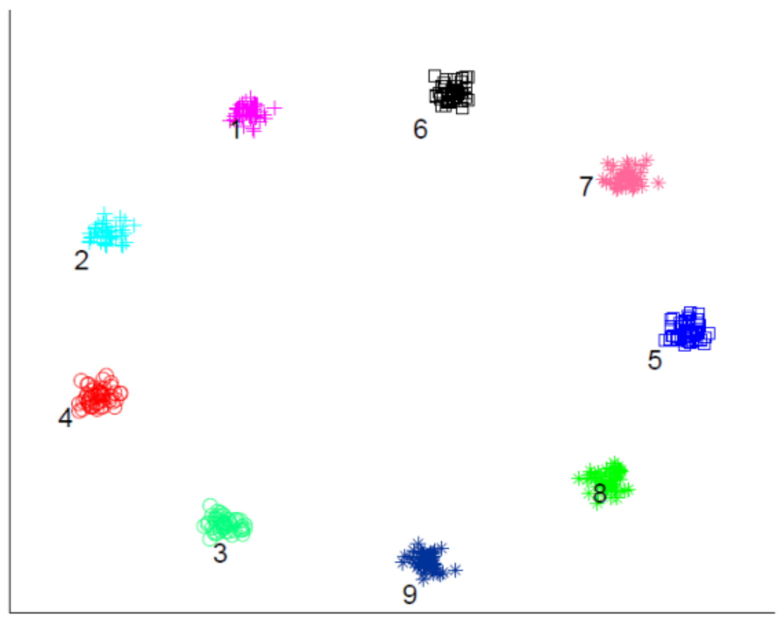
5-gram	Weight
is an extremely good book	3.58
is just a good book	3.19
book is a good buy	2.94
overall a very good book	2.84
book is a good choice	2.81
book is a very good	2.76
book is still very good	1.70
a good book just because	1.15
unless good books are just	1.05

Sentiment Classification with Supervised Sequence Embedding; D. Bespalov et al. ECML 2012

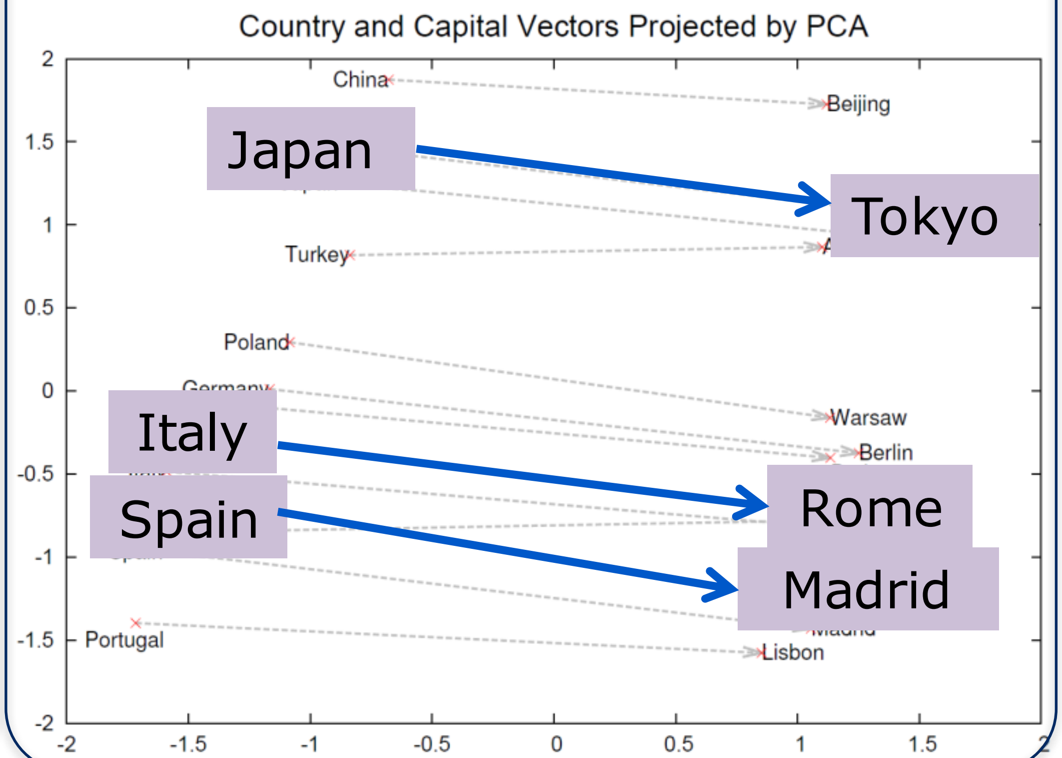
Semantic Embeddings

Map words or concepts into a vector space
Semantically similar words/concepts should be close.
Very effective for recommender systems and other applications

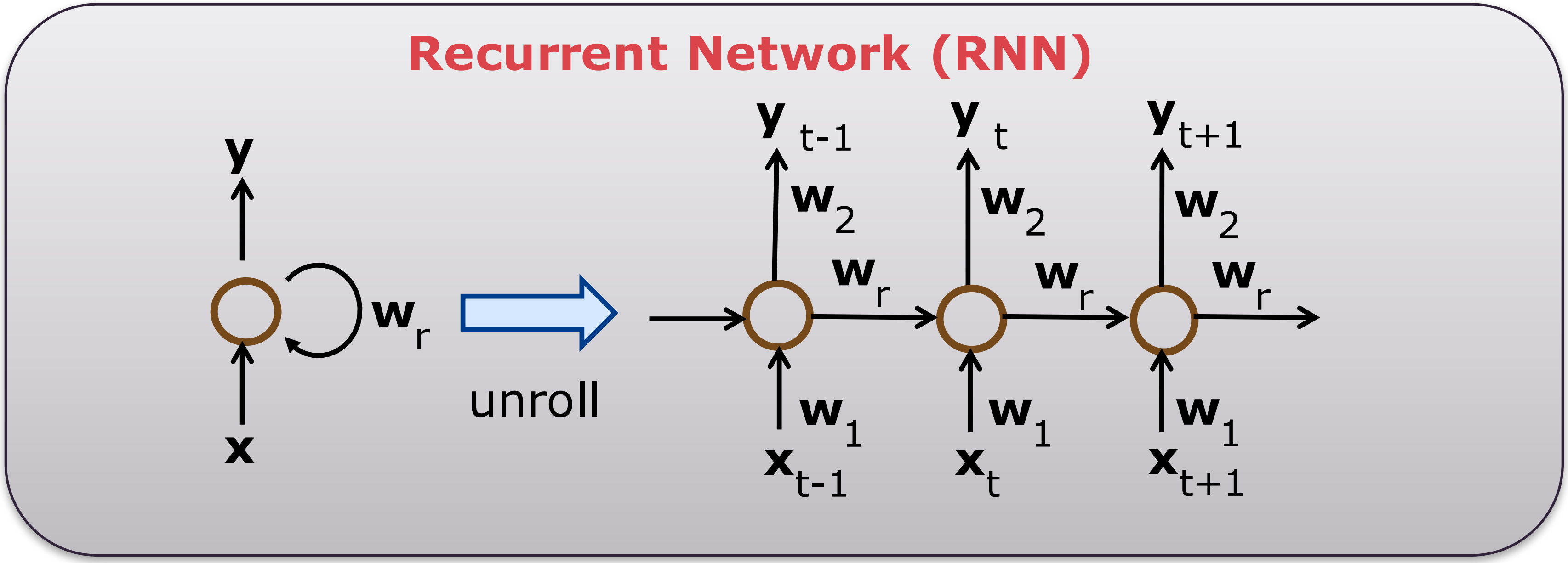
Example: Embedding of articles presented at NIPS. Papers with similar topics are close together in vector space (Deep Semantic Embedding)



Example: Word2Vec
Relations like Country – Capital can be visualized in vector Embedding space.



Learning Sequences with Recurrent Networks



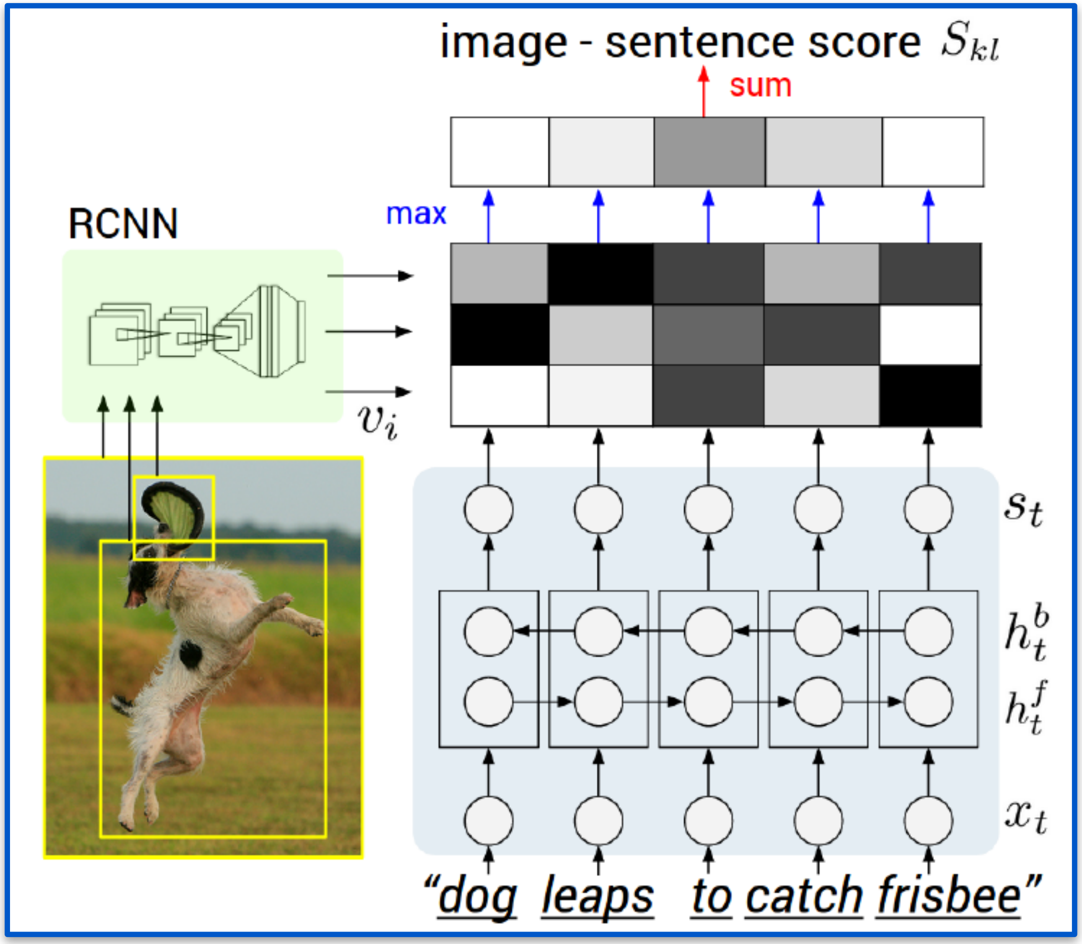
Sequence learning is a bit tricky, since the network may not understand well, how long to keep old information.
Add gates to make it learn better, e.g. **Long Short Term Memory** (LSTM)

Other attempts to Sequence Learning are **Memory Networks**:
- Store embeddings of information in memory locations.
- Learn how to retrieve relevant information.

Torch code for example of Memory Networks:
https://github.com/facebook/MemNN

Generating Image Captions Automatically

Goal: Given an image, generate a caption that summarizes concisely the content of the image. Requires semantic interpretation of image. An old AI problem that has made little progress over the last decades.



- A deep learning CNN analyzes the image to summarize image elements, and produce semantic embedding.
- Output of CNN is entered into a Recurrent Neural Net (RNN) for generation of the text.

[1]: A. Karpathy, FeiFei Li, CVPR 2015

Torch code: https://github.com/karpathy/neuraltalk2

Generating Image Captions Automatically: Examples

Examples from Ref. 1

Human Label	bowls are food in triangular shape are sitting on table	hotdog stand on busy street
Closest training sample	table filled with many plates of various breakfast foods	man in white t shirt is holding umbrella and ice cream cart
Text generated by RNN	table topped with lots of different types of donuts	man in white shirt is pushing his cart down street

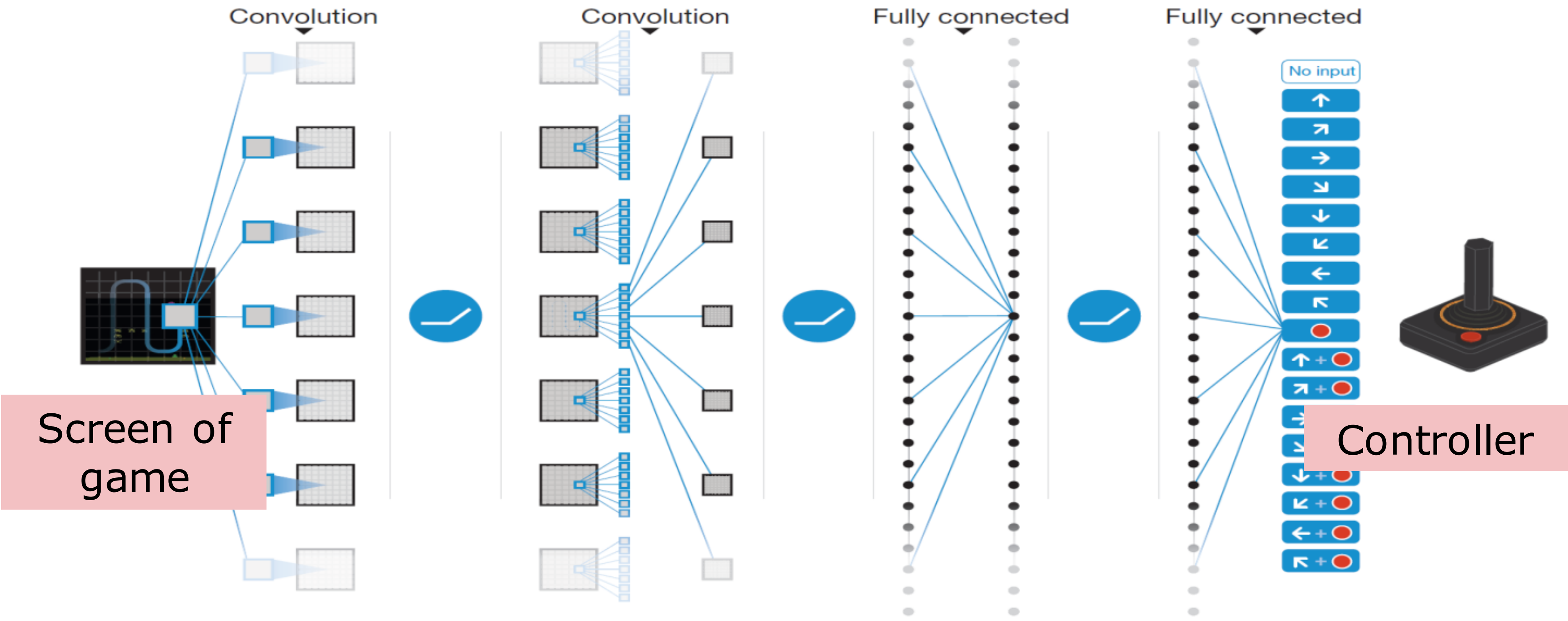
Tests with **Neuraltalk2**:
Images from our lab

A group of people playing Frisbee in a field.

A group of people standing around a table with a cake

Learning to Play Video Games

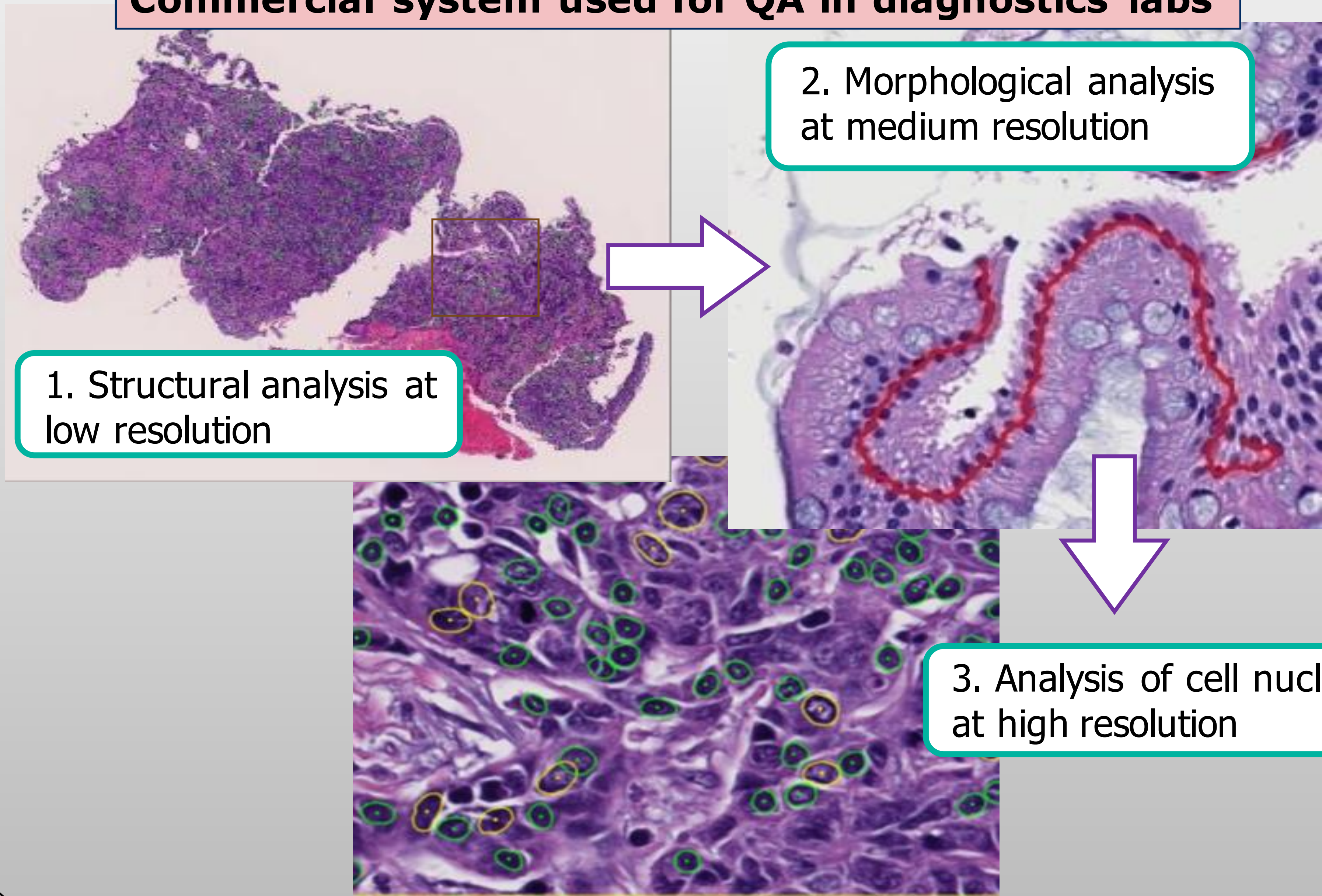
Deep-Q Network (Google DeepMind): Learn to play video game. Look at screen and get rewards for good moves (Reinforcement Learning). From screen image → Automatically generate commands for Controller



Human-level control through deep reinforcement learning; V. Mnih, K. Kavukcuoglu et al.
NATURE: 6 FEBRUARY 2015 | VOL 518 | p 529
Torch code: https://github.com/kuz/DeepMind-Atari-Deep-Q-Learner

Medical Applications: Digital Pathology

Commercial system used for QA in diagnostics labs



Is a Deep Learning Net the next US President?

Twitterbot created by B. Hayes, CSAIL
https://www.csail.mit.edu/deepdrumpf

- Trained with text spoken by **Donald Trump**

DeepDrumpf
@DeepDrumpf

#MakeLSTMGreatAgain #MakeAmericaLearnAgain I'm a Neural Network trained on Donald Trump transcripts. (Priming text in [ ]s). Follow … tails.

DeepDrumpf @DeepDrumpf · 16h
Mark my words. We're going to beat ISIS. Come replace the big lie, Obamacare. Believe me.

DeepDrumpf @DeepDrumpf · 21h
Right now, think of this: We owe China $1.3 trillion. We owe Japan more than that. We have gun laws. I'll bring back our money.

Parallelization of Deep Learning

Large efforts under way, developing parallel implementations
Several types of parallelization need to be considered

Parallelization	Software	Characteristics	Speed
Multi-Core	• MKL • BLAS	• Shared memory of cores Compiler optimized	• Limited performance, but very flexible.
Cluster, Many core	• OpenMP, …	• Communication over high-speed network; message passing	• Potential for high performance, but usually limited to few processors. • Challenge: Data or Model parallelization over a network
Vector	• Intel: AVX512	• SIMD, Vector unit integrated with CPU.	• Good performance per core • Challenge: Mixture of regular and vectorized operations
	• NEC: SX	• SIMD, wide registers; Coprocessor	• High performance per core. Challenge: Vectorize all layers
GPU	• CudaNet • CuDNN	• Up to 3,000 cores • Coprocessor	• High raw speed. • Challenge: Use all cores efficiently. Often restrictions on functionality.
FPGA	Specialized instruction set.	• SIMD, Systolic, … • Specialized instruction set	• Good for special functionality. Challenge: Low clock frequency.

Benchmarks of Open Source Systems

https://github.com/soumith/convnet-benchmarks; Accessed **March 5, 2016**

Machine: 6-core Intel Core i7-5930K CPU @ 3.50GHz + **NVIDIA Titan X** + Ubuntu 14.04 x86_64

OxfordNet [Model-A] - Input 64x3x224x224

Library	Class	Time (ms)	forward (ms)	backward (ms)
Nervana-neon-fp16	ConvLayer	254	82	171
Nervana-neon-fp32	ConvLayer	320	103	217
CuDNN[R4]-fp16 (Torch)	cudnn.SpatialConvolution	471	140	331
CuDNN[R4]-fp32 (Torch)	cudnn.SpatialConvolution	529	162	366
Chainer	Convolution2D	885	251	632
TensorFlow	conv2d	982	191	791
fbfft (Torch)	SpatialConvolutionCuFFT	1092	355	737
cudaconvnet2*	ConvLayer	1229	408	821
CuDNN[R2] *	cudnn.SpatialConvolution	1099	342	757
Caffe	ConvolutionLayer	1068	323	745
Torch-7 (native)	SpatialConvolutionMM	1105	350	755
CL-nn (Torch)	SpatialConvolutionMM	3437	875	2562
Caffe-CLGreenTea	ConvolutionLayer	5620	988	4632

Winograd FFT

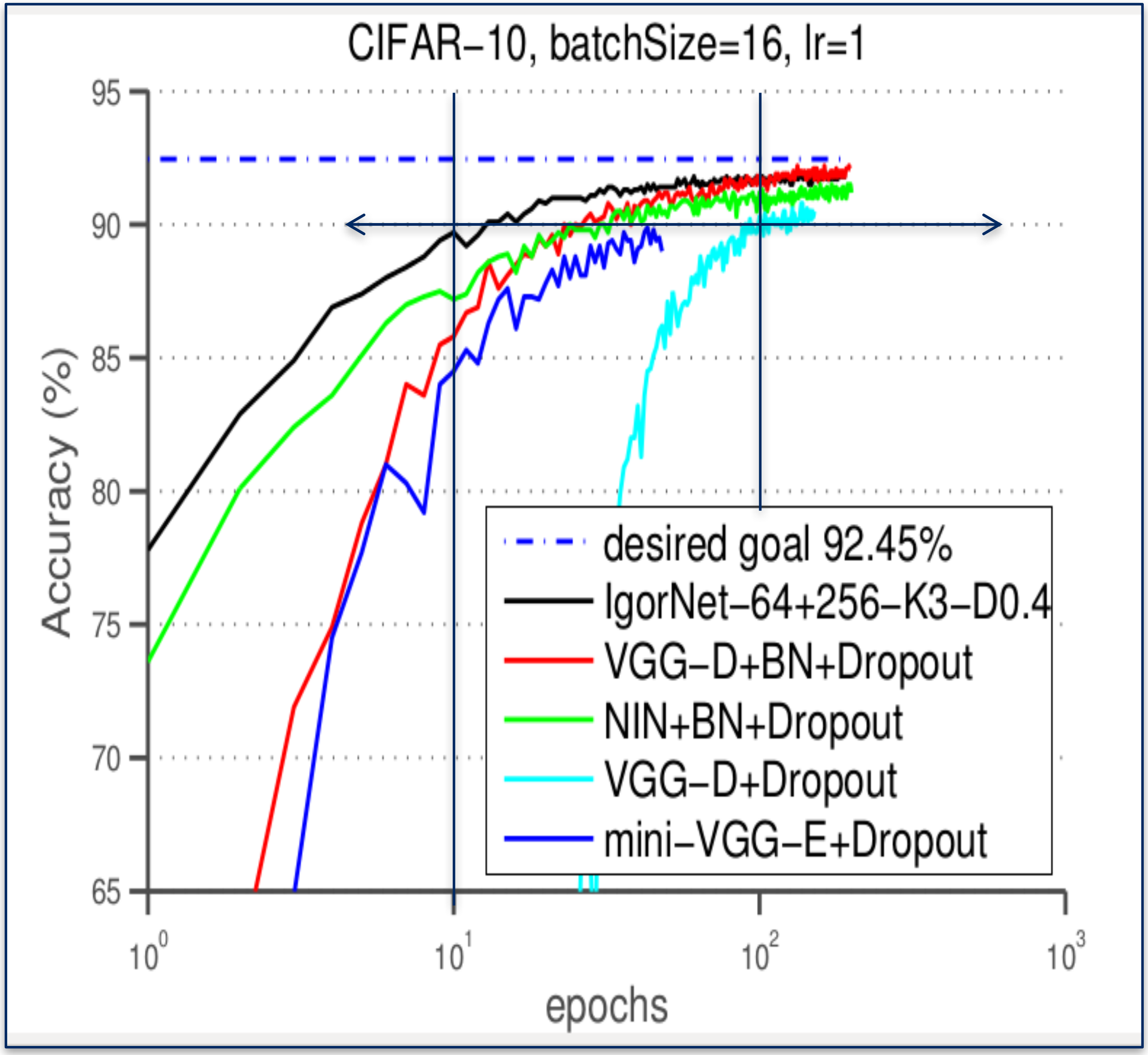
Benchmarking Deep Learning Networks

Efficiency: Which network learns fastest (with fewest samples)?
GPU may lack flexibility to implement features that make learning fast.

Example of training with various networks:
Compare number of epochs to get to 90% accuracy

Compute time **varies 7x to >17x among various** networks that have best accuracies, depending on what features are activated

Measured with Torch 7



Conclusions

Data analytics and AI are very dynamic fields and making fast progress.
Need environment that can handle a wide range of problems

Deep Learning is the hottest topic today, but what is tomorrow?

Research is moving beyond multi-layer networks.

Reasoning encompasses much more than pattern matching.

Torch 7:

- **Flexibility to handle any data type: Speech, Text, Image, Video, Time series, Machine data, Sensor data, ...**
- **Active Open Source eco system**
- **Very efficient: Designed for performance from the ground up**
- **Tools to connect to other frameworks**
- **Runs on any platform: From data centers to embedded devices.**
- **Industry support: API's from Intel, NVIDIA, AMD, ...**

Orchestrating a brighter world

www.nec-labs.com