UNIVERSITY of HOUSTON

DEPARTMENT OF PHYSICS

Deep Learning

Zhihua Liang

At UCI, 02/18/2015

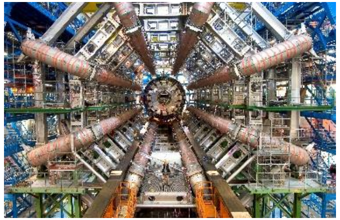
About me

SMU

ATLAS

01

2006



C T E Q

02

2012

SMU

Parton Distribution-

Function

MEKS(GPU)

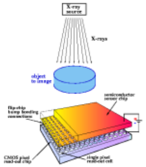
UH

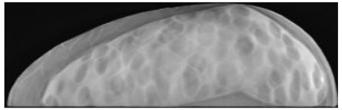
X-ray Simulation

Medipix

03

2013





04

2014

UH

Lesion detection

Machine Learning

?

05

2015

About me

ATLAS

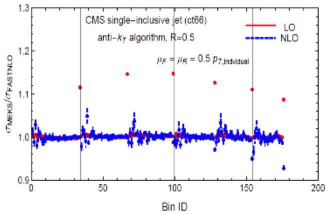
GEANT4
ROOT
COOL
PANDA
C++
QT4
Python

CTEQ

MEKS:

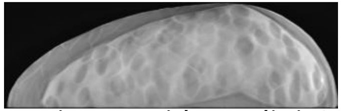
a program computes differential cross-section for single-inclusive jets and di-jet pairs at NLO accuracy in pQCD

EKSMU: a version of MEKS that runs on GPU.



Medical/Image Physics

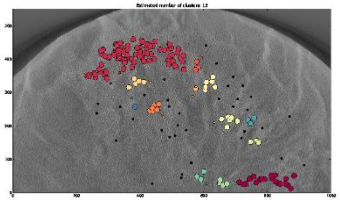
A c++ code to simulate x ray attenuation/phase contrast image



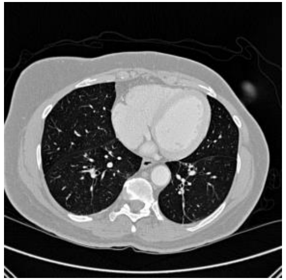
Experiment with Medipix3, a photon courting detector.

Machine Learning

A Model Observer using Histogram of Gradients(HOG), Support Vector Machine(SVM), and Density-based spatial clustering of applications with noise (DBSCAN) methods.



A Lung lesion detector with CNN.



01 2006

02 2012

03 2013

04 2014

05 2015

Outline

1/ Introduction

2/ (Un-)Supervised Learning

3/ Practical Tools

4/ Deep Learning in HEP

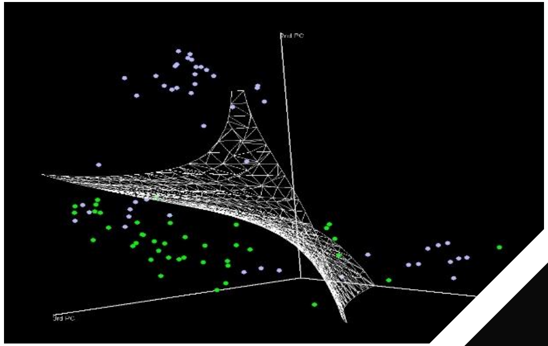
Learning Methods

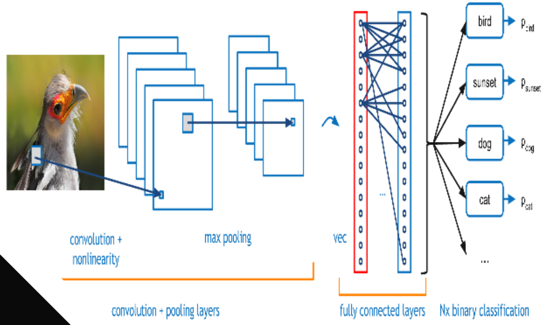
Supervised

Supper Vector Machine

Logistic Regression

Perceptron





Deep Neural Net

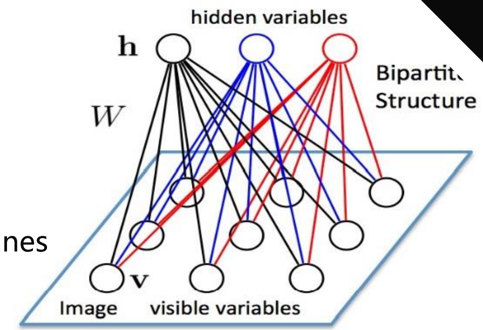
Convolutional Neural Net

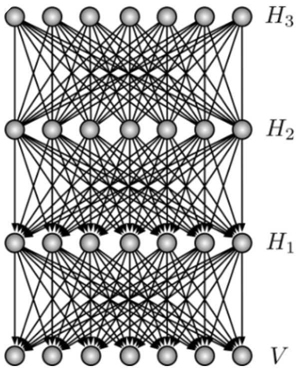
Recurrent Neural Net

Shallow

Machine Learning

Deep





Denoising Autoencoder

Restricted Boltzmann Machines

Sparse Coding

Deep Belief Nets

Deep Boltzmann Machines

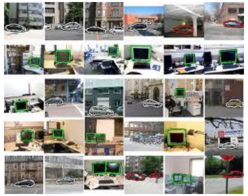
Hierarchical Sparse Coding

Unsupervised

ML Applications

Image and Videos

flickr
Google
YouTube

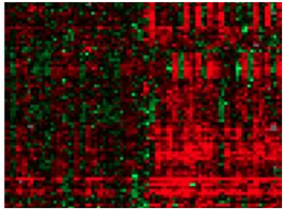


Text and Language

Wikipedia
The Free Encyclopedia
REUTERS
AP Associated Press

Speech and Audio

Gene Expression

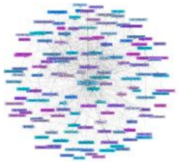


Product Recommendation

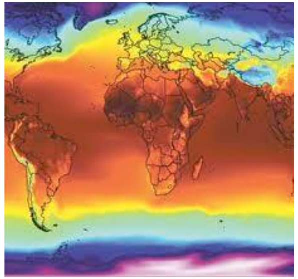
ebay
amazon
NETFLIX

Social Network

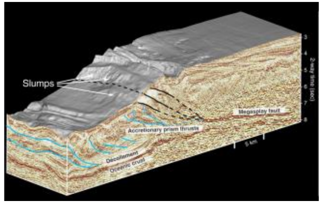
facebook
twitter



Climate Change



Geology



Traditional Recognition Approach

Low-level sensing

feature extraction & selection
(hand-crafted)

Low-level
vision features
(edges, SIFT, HOG, etc.)

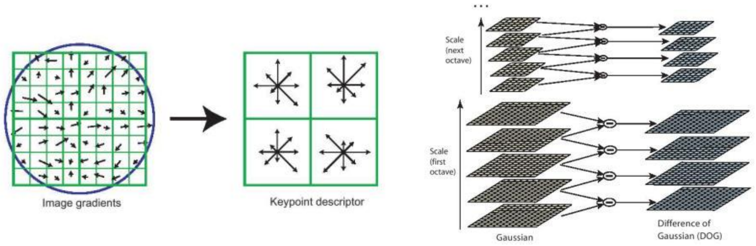
- Most critical for accuracy
- Account for most of the computation for testing
- Most time-consuming in development cycle
- Often hand-craft in practice

Object detection / classification

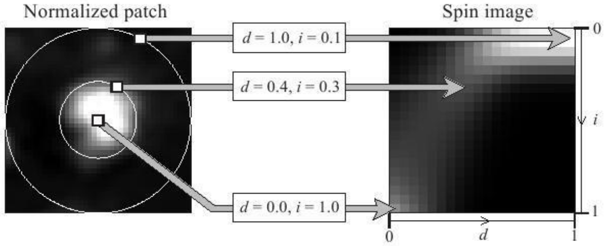
Learning
Algorithm
(e.g., SVM)

Computer vision features

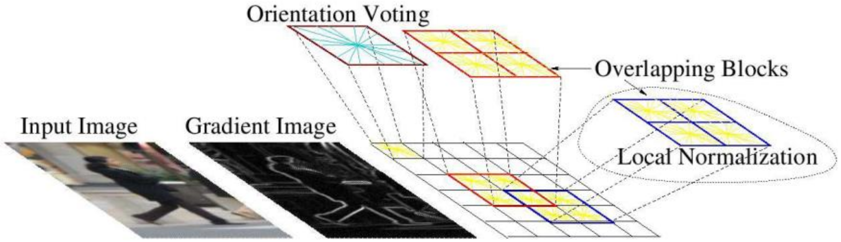
SIFT



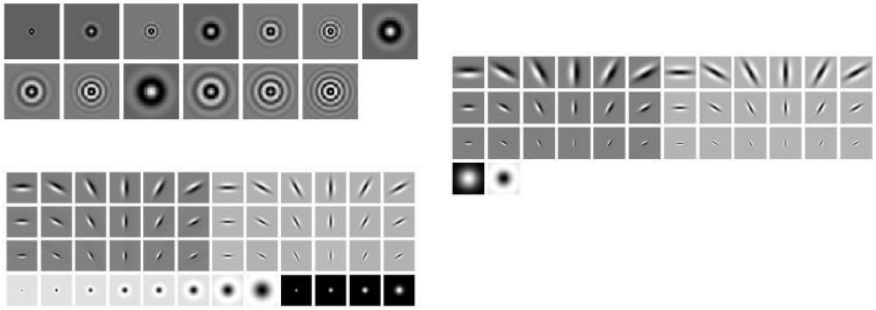
Spin Image



HoG



Textons



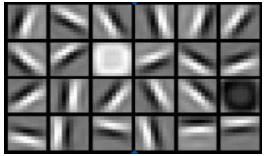
and many others:
SURF, MSER, LBP, Color-SIFT, Color histogram, GLOH, …..

Learning Feature Hierarchy

Feature representation

3rd layer
Objects

2nd layer
Objects



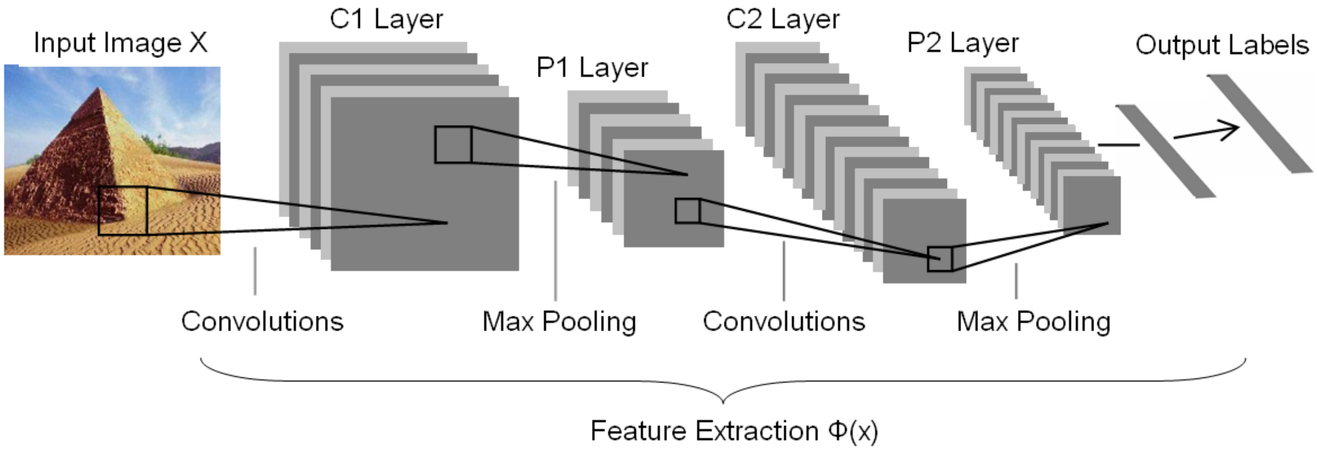
1st layer
Edges

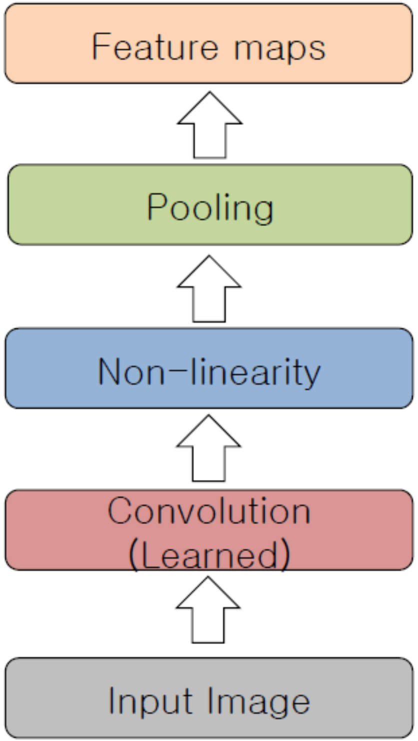
Pixels

Learn hierarchy

- All the way from pixels → classifier
- One layer extracts features from output of previous layer
- Learn useful higher-level features from images
- Fill in representation gap in recognition

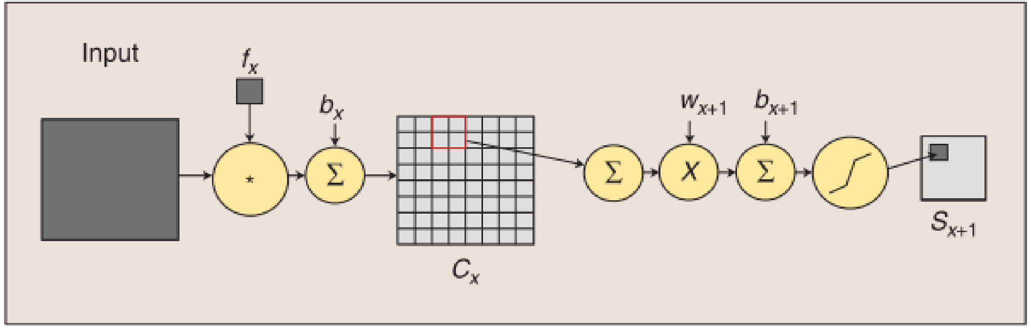
Convolution Neural Networks (CNNs): 1989



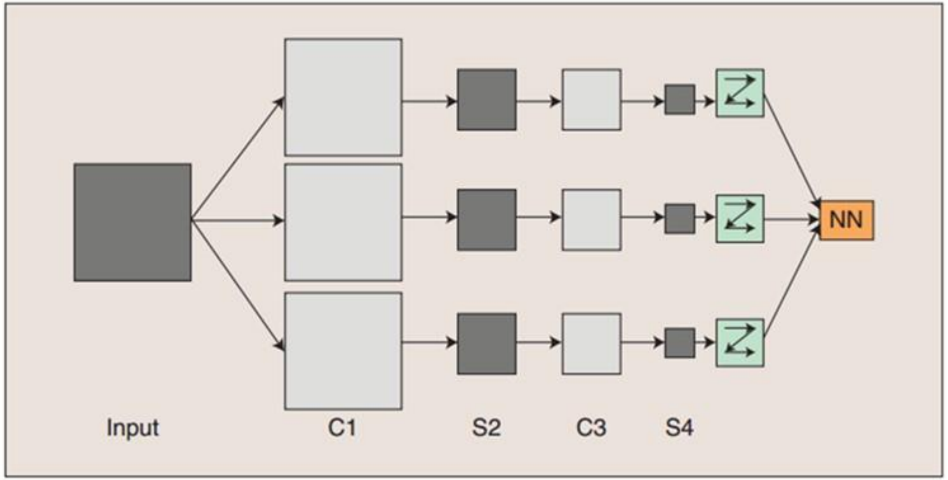


LeNet: a layered model composed of convolution and subsampling operations followed by a holistic representation and ultimately a classifier for handwritten digits.

Convolution Neural Networks (CNNs): 1989

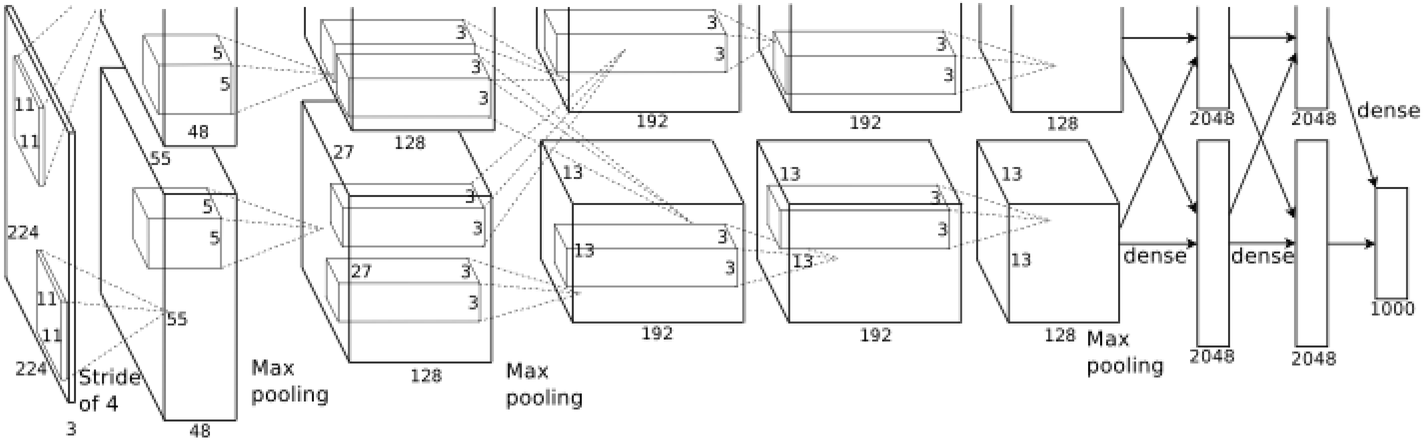


The input is converted into a convolution layer Cx via the convolution process. Then, Cx is treated as the input data, converted to a set of smaller-dimension feature maps Sx+1 via the subsampling process



An entire procedure of using CNNs to compute a output. We can see that after each layer, the dimensions of the feature maps are decreased, which makes CNNs a useful model to reduce the number of parameters that must be learned and thus improves
upon general feed-forward back-propogation training.

Convolutional Nets: 2012

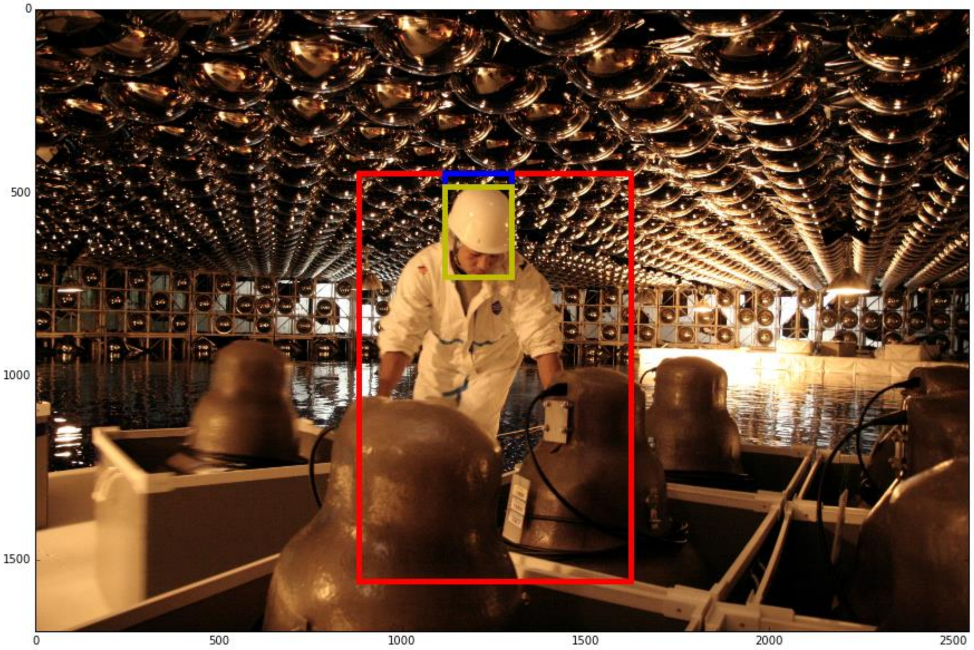


AlexNet: a layered model composed of convolution, subsampling, and further operations followed by a holistic representation and all-in-all a landmark classifier on ILSVRC12

+ data
+ gpu
+ non-saturating nonlinearity
+ regularization (DropOut)

7 hidden layers, 60 M parameters, Trained on 2 GPUs for a week

R-CNN: CNN on detection



Top detection:

person	2.467470
swimming trunks	-1.263134
puck	-1.296530

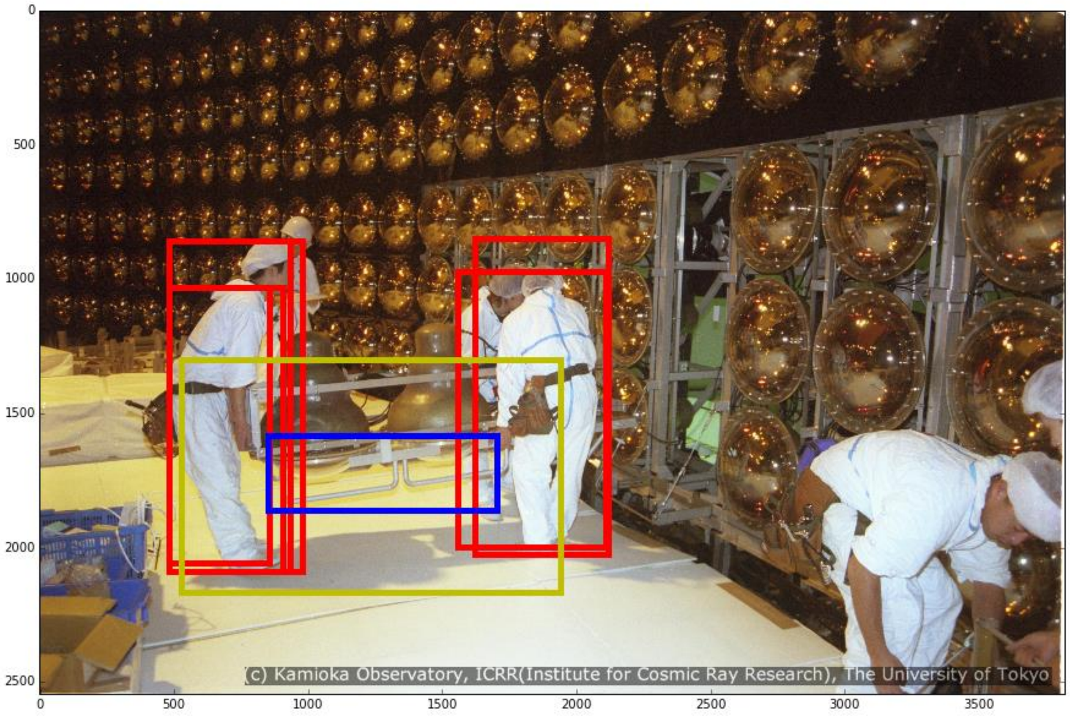
Second-best detection:

mushroom	0.015308
helmet	-0.656349
bathing cap	-0.736946

Third-best detection:

mushroom	-0.210073
helmet	-0.268050

R-CNN: CNN on detection



Top detection:

person	3.855254
bow	-1.046176
chair	-1.302708

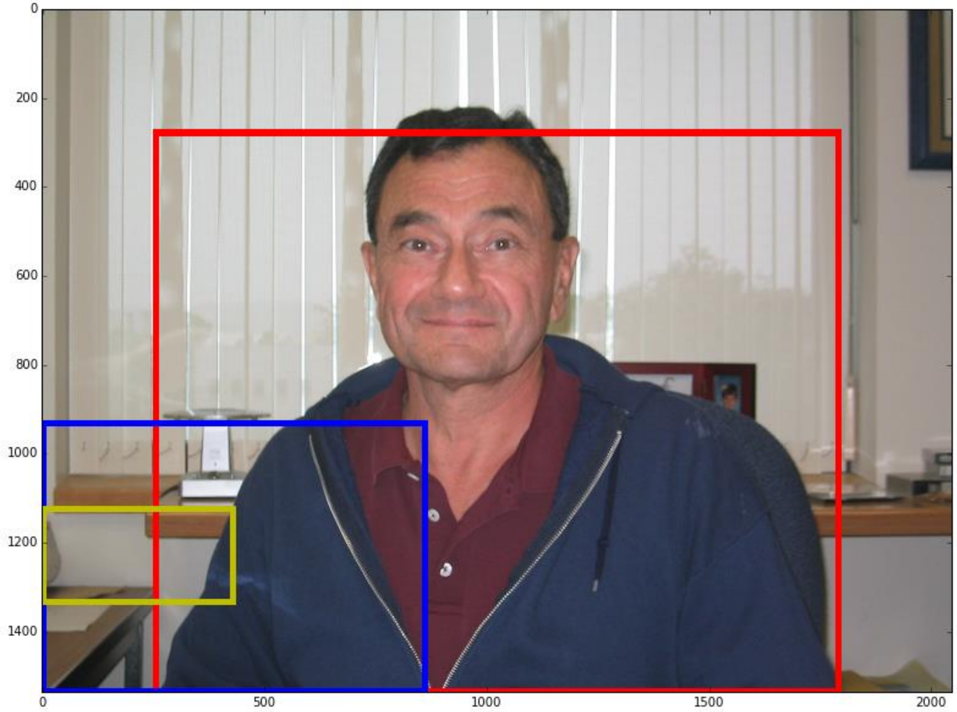
Second-best detection:

screwdriver	-0.177056
vacuum	-0.951643
spatula	-0.971048

Third-best detection:

stove	-0.325801
snowplow	-0.580681
toaster	-0.746113

R-CNN: CNN on detection



Top detection:

person	1.812851
snowplow	-1.222213
bow	-1.271971

Second-best detection:

table	-0.782973
dishwasher	-1.269662
golf ball	-1.279250

Third-best detection:

sofa	-0.951138
bench	-1.278376
spatula	-1.534883

R-CNN: CNN on detection

Top detection:

person 1.227990
purse -1.180718
flower pot -1.322483

Second-best detection:

helmet -0.383887
person -1.007048
bicycle -1.079069

Third-best detection:

water bottle -0.849368
cup or mug -1.121394
dumbbell -1.203280

R-CNN: CNN on detection

Top detection:

person	2.105693
purse	-1.266800
binder	-1.387294

Second-best detection:

diaper	-0.342870
plastic bag	-0.784050
backpack	-1.197001

Third-best detection:

miniskirt	-0.354545
band aid	-1.249658
syringe	-1.321468

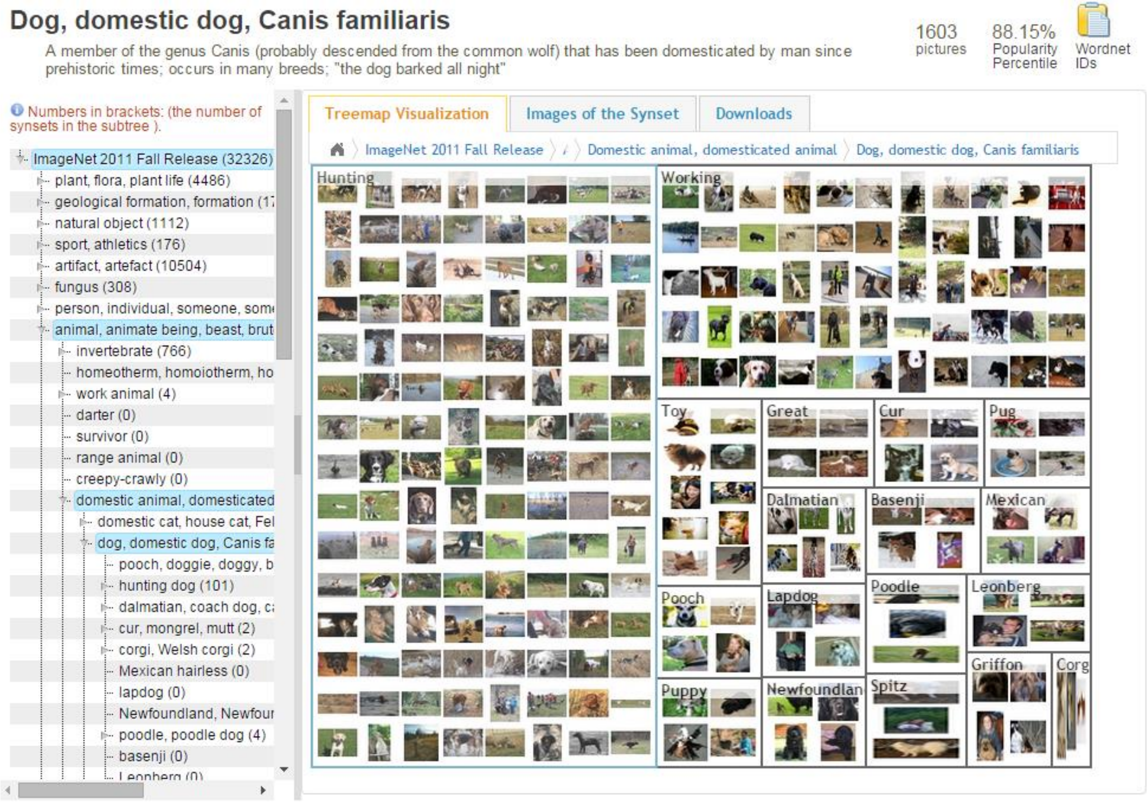
ILSVRC12

Team name	Error (5 guesses)	Description
SuperVision	0.15315	AlexNet
ISI	0.26172	Weighted sum of scores from each classifier with SIFT+FV, LBP+FV, GIST+FV, and CSIFT+FV, respectively.
OXFORD_VGG	0.26979	Mixed selection from High-Level SVM scores and Baseline Scores

Krizhevsky et al. -- 16.4% error (top-5)
Next best (non-convnet) – 26.2% error

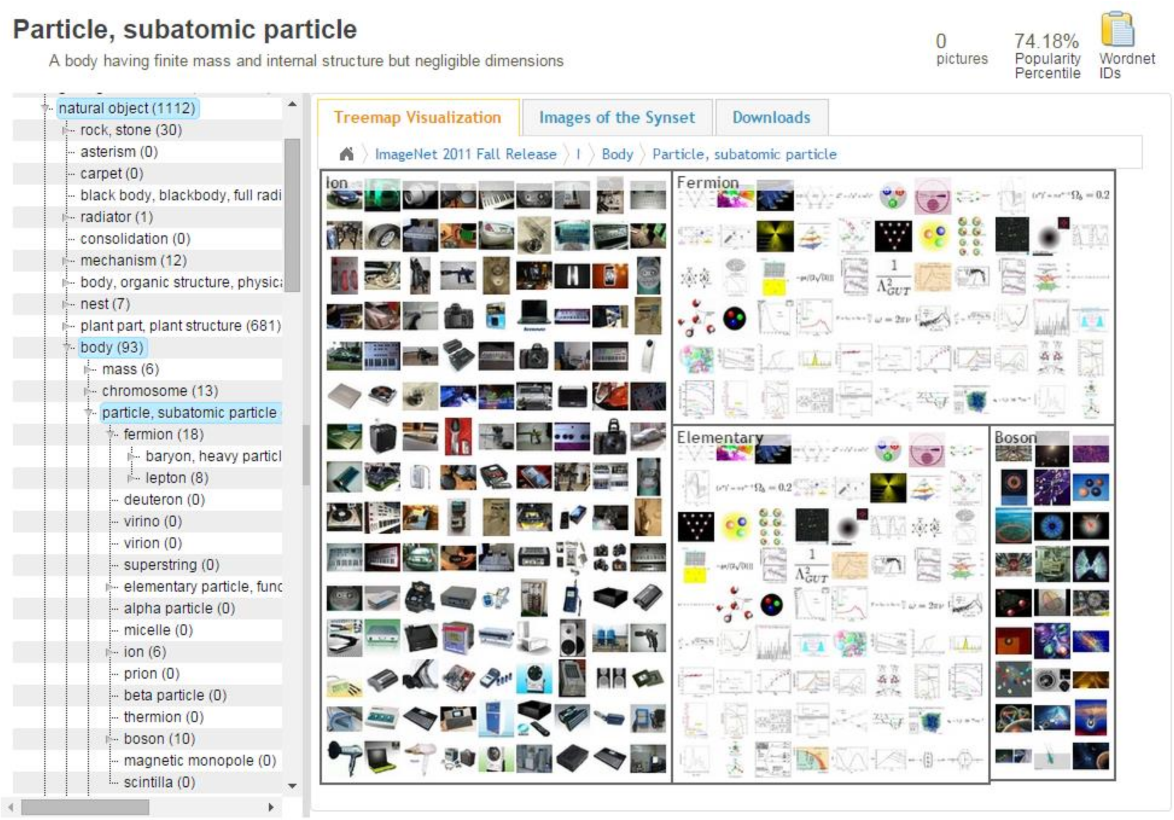
The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) evaluates algorithms for object detection and image classification at large scale.
ImageNet is an image database with 14,197,122 labeled images, 20k classes

ImageNet



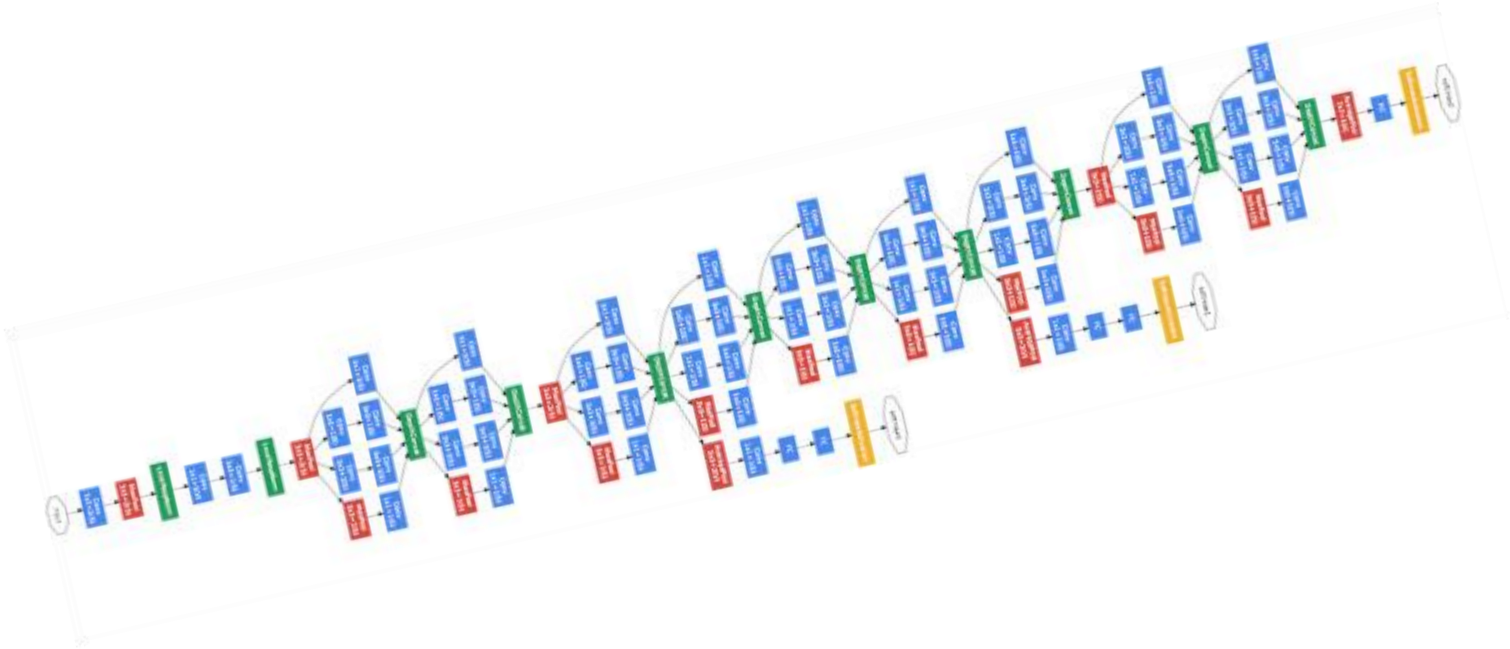
ImageNet category of Dog, domestic dog, Canis familiaris

ImageNet



ImageNet category of Particles….

Convolutional Nets: 2014

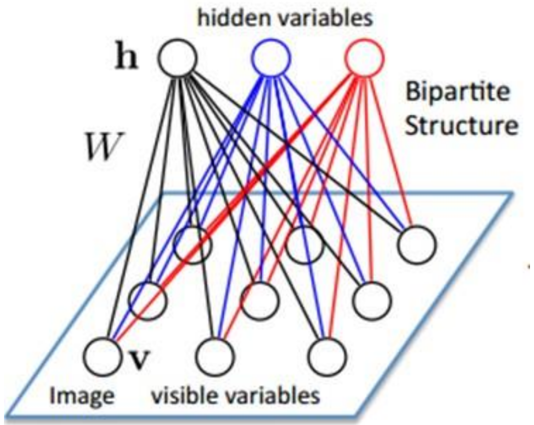


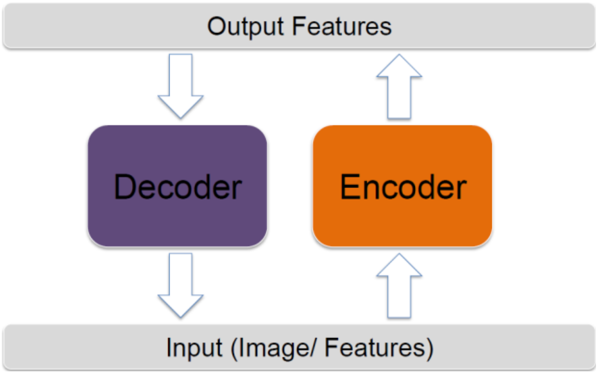
ILSVRC14 Winners: **~6.6% Top-5 error**

- GoogLeNet: composition of multi-scale dimension-reduced modules (pictured)
- VGG: 16 layers of 3x3 convolution interleaved with max pooling + 3 fully-connected layers

+ depth
+ data
+ dimensionality reduction

Unsupervised Learning

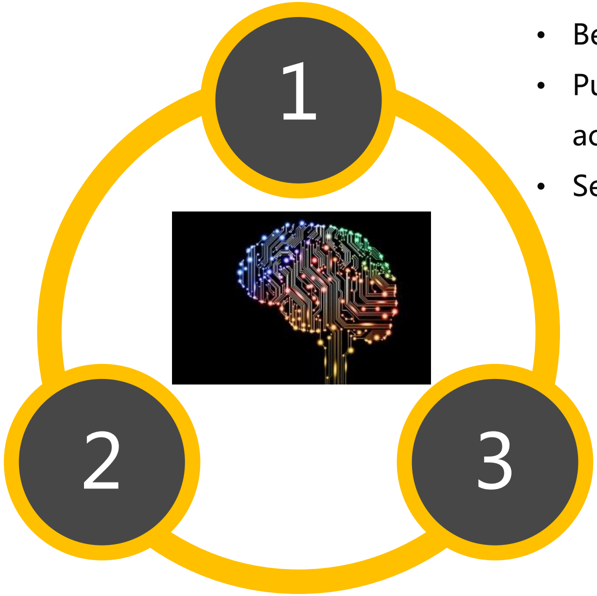




- Model distribution of input data
- Can use unlabeled data (unlimited)
- Can be refined with standard supervised techniques (e.g. backprop)
- Useful when the amount of labels is small

Models
- Basic: PCA, KMeans
- Denoising auto-encoders
- Sparse auto-encoders
- Restricted Boltzmann machines
- Sparse coding
- Independent Component Analysis
- ...

Deep Learning Libraries



Caffe

- Berkeley Vision and Learning Center (BVLC)
- Pure C++ / CUDA architecture with CuDNN acceleration
- Seamless switch between CPU and GPU

Torch7

- NYU
- scientific computing framework in Lua
- supported by Facebook

Theano/Pylearn2

- U. Montreal
- scientific computing framework in Python
- symbolic computation and automatic differentiation

All express deep models

All are nicely open-source

All include scripting for hacking and prototyping

Caffe

- Speed with Krizhevsky's 2012 model:
  - K40 / Titan: 2 ms / image, K20: 2.6ms
  - Caffe + cuDNN: **1.17ms / image** on K40
  - 60 million images / day
  - 8-core CPU: ~20 ms/image
- ~ 9K lines of C/C++ code
  - with unit tests ~20k

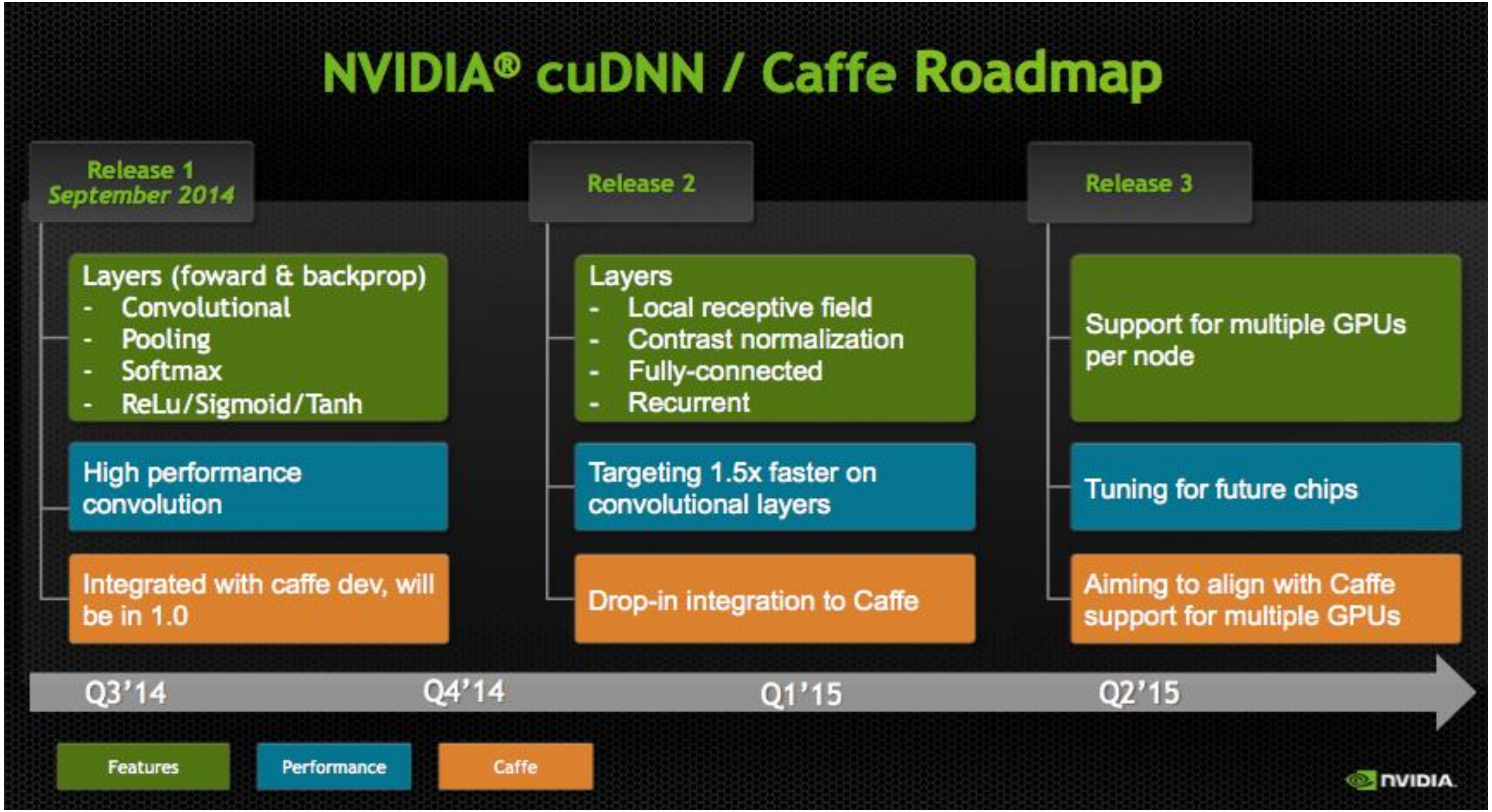
Caffe offers the
model definitions
optimization settings
pre-trained weights
so we can start right away.

The BVLC models are
licensed for unrestricted use.

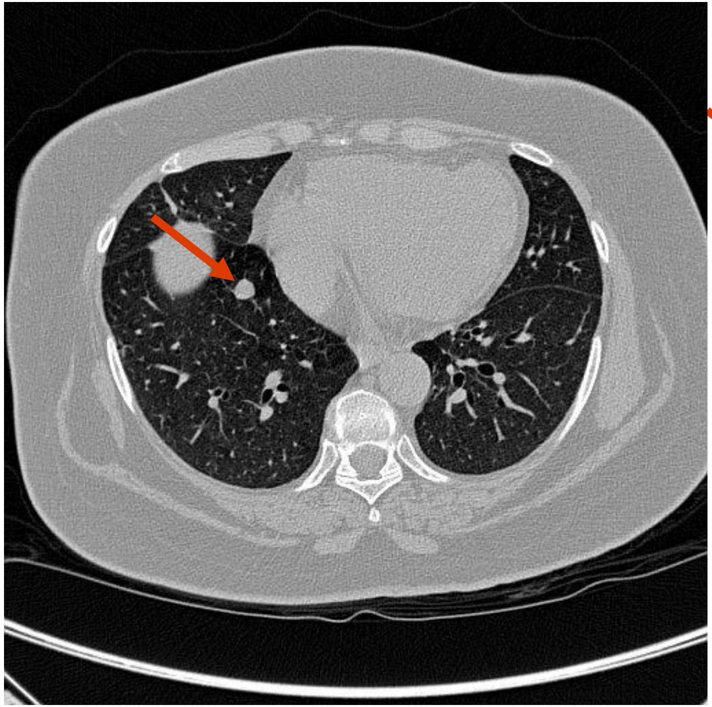
C++ 84.2% Python 10.5% Cuda 3.9% Other 1.4%

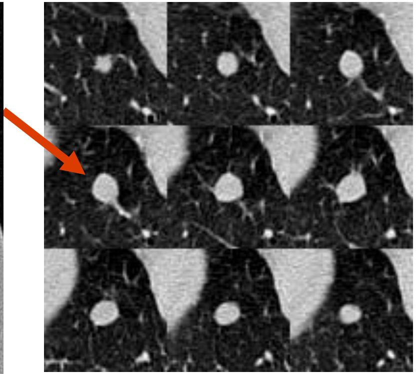
NVIDIA cuDNN is a GPU-accelerated library of primitives for deep neural networks.

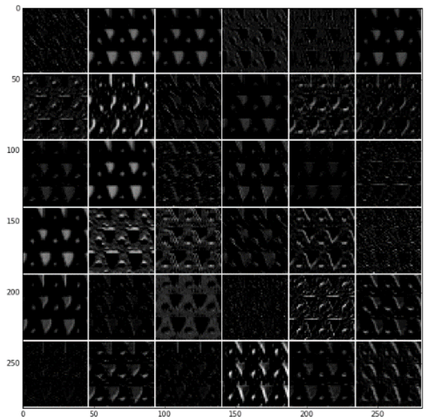
Caffe/CuDNN roadmap



A 3D lesion detector







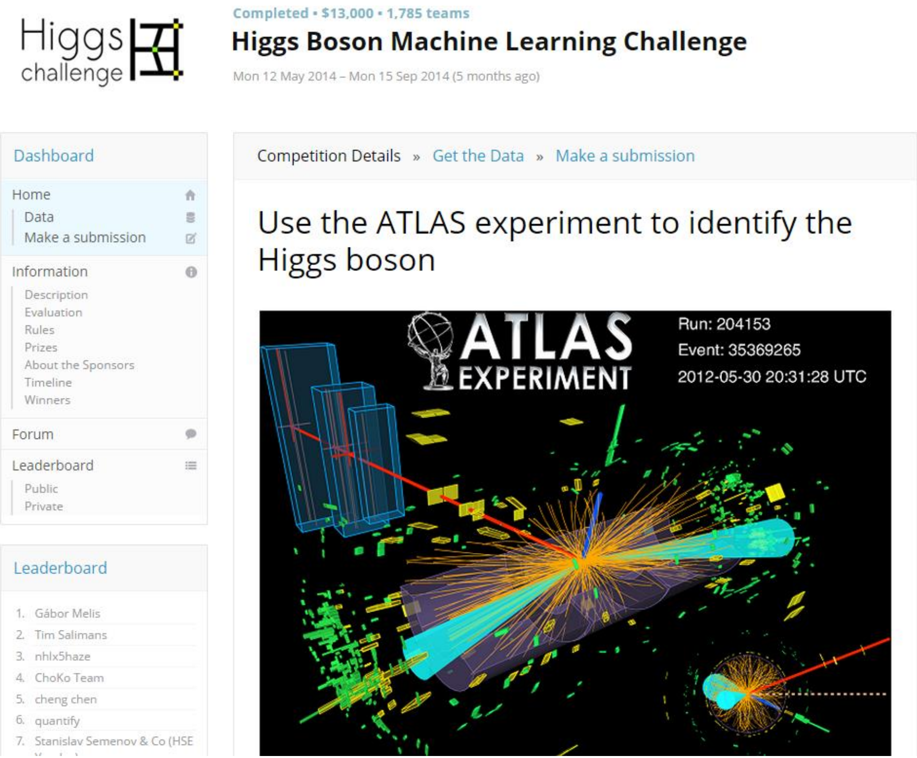
A 3-D lesion detector for Lung tumor is developed based on caffe.

Adopted CNN network but works on CT images(3D, gray color)

Trained on Lung Image Database Consortium (LIDC) dataset. (1012 cases, each case is a 512*512*(200-400) image.

Rank 4/14 on SPIE Lung Nodule Classification Challenge with training on only 100 labeled cases. (would be better if we have more labeled cases, or have pre-trained with unsupervised learning then fine-tune with labeled data)

Higgs Boson Machine Learning Challenge



Winning submission:

Bag of 70 dropout (50%) deep neural networks (600,600,600), channel-out activation function, l1=5e-6 and l2=5e-5 (for first weight matrix only). Each input neurons is connected to only 10 input values (sampled with replacement before training).

Higgs: Binary Classification Problem

UCI
Machine Learning Repository
Center for Machine Learning and Intelligent Systems

HIGGS Data Set

Download: Data Folder, Data Set Description

Abstract: This is a classification problem to distinguish between a signal process which produces Higgs bosons and a background process which does not.

Data Set Characteristics:	N/A	**Number of Instances:**	11000000	**Area:**	Physical
Attribute Characteristics:	Real	**Number of Attributes:**	28	**Date Donated**	2014-02-12
Associated Tasks:	Classification	**Missing Values?**	N/A	**Number of Web Hits:**	17738

HIGGS UCI Dataset:
21 low-level features AND
7 high-level derived features
Train: 10M rows, Test: 500k rows

nature COMMUNICATIONS

Home | About the journal | Authors and referees | Browse archive | Search

nature.com ▸ journal home ▸ archive by date ▸ july ▸ abstract

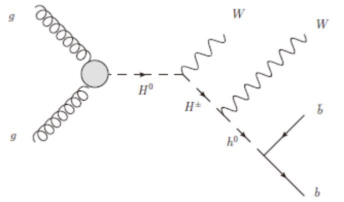
NATURE COMMUNICATIONS | ARTICLE

Searching for exotic particles in high-energy physics with deep learning

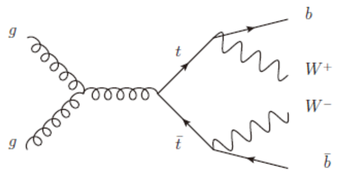
P. Baldi, P. Sadowski & D. Whiteson

Affiliations | Contributions | Corresponding authors

Nature Communications **5**, Article number: 4308 | doi:10.1038/ncomms5308
Received 19 February 2014 | Accepted 04 June 2014 | Published 02 July 2014

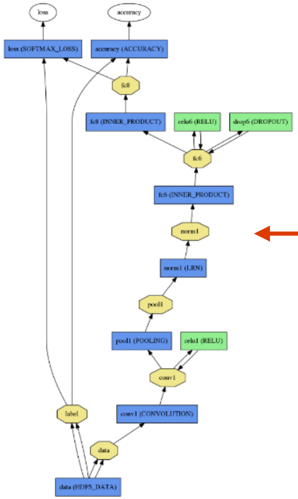


(a)

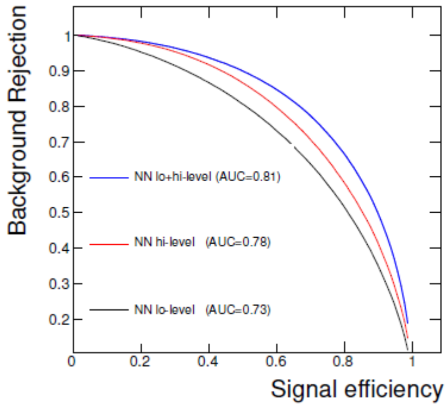


Higgs: Binary Classification Problem

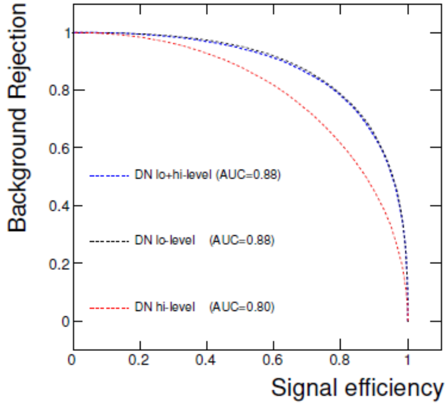
AUC			
Technique	Low-level	Low-level	Complete
BDT	0.73	0.78	0.81
NN	0:733	0.777	0.816
DN	0.880	0.800	0.855



Accuracy = 0.73 with this network seting



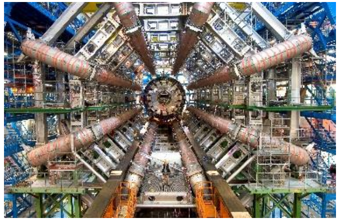
(a)

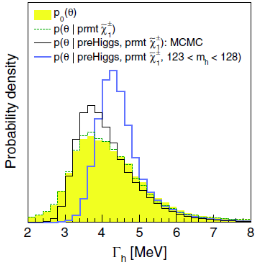


Can we benefit from this deep learning revolution?

Traditional computer vision task processing

Low-level sensing → Pre-processing → Feature extract. → Feature selection → Inference: prediction, recognition





Examine signal and background properties, set "cuts"

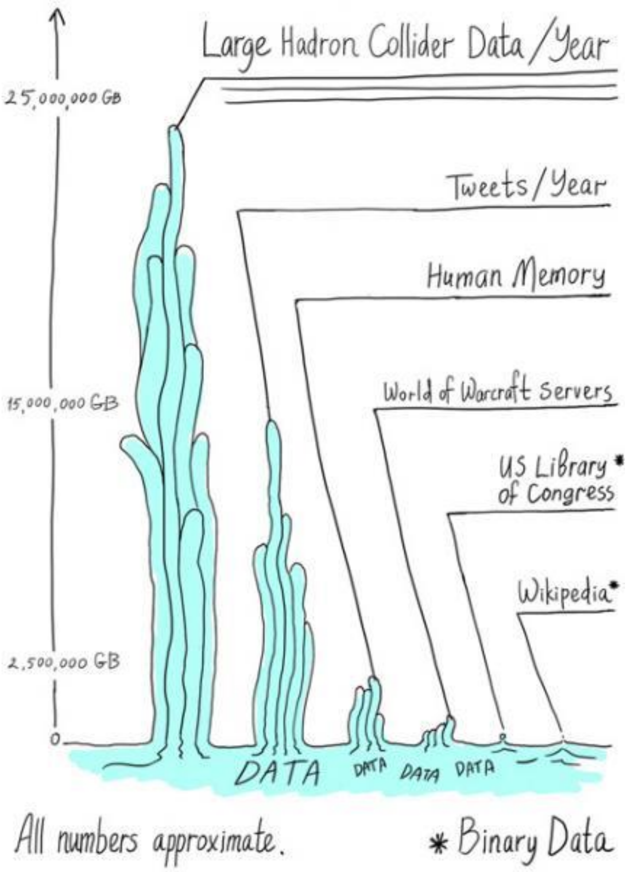
Result

HEP

Thank You!

Zhihua Liang, 02/18/2015

Big Data



LHC: 25 petabytes

Youtube: One hour video are uploaded per second…
=3600*24*365*1GB=31 petabytes?

Logistic Regression

The Model

Classification is done by projecting an input vector onto a set

$$P(Y=i|x,W,b) = softmax_i(Wx+b) = \frac{e^{W_i x + b_i}}{\sum_j e^{W_j x + b_j}}$$

$$y_{pred} = \mathrm{argmax}_i P(Y=i|x,W,b)$$

Loss Function

Learning optimal model parameters involves minimizing a loss function.

$$\mathcal{L}(\theta=\{W,b\},\mathcal{D}) = \sum_{i=0}^{|\mathcal{D}|} \log(P(Y=y^{(i)}|x^{(i)},W,b))$$

$$\ell(\theta=\{W,b\},\mathcal{D}) = -\mathcal{L}(\theta=\{W,b\},\mathcal{D})$$

Caffe net plot

Multilayer Perceptron

The Model

Classification is done by projecting an input vector onto a set

$$f(x) = G(b^{(2)} + W^{(2)}(s(b^{(1)} + W^{(1)}x))),$$

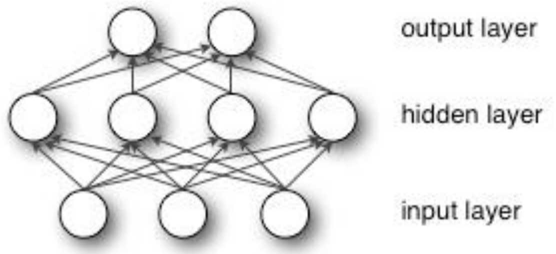
$$y_{pred} = \mathrm{argmax}_i P(Y = i|x, W, b)$$

Loss Function

Learning optimal model parameters involves minimizing a loss function.

$$\mathcal{L}(\theta = \{W, b\}, \mathcal{D}) = \sum_{i=0}^{|\mathcal{D}|} \log(P(Y = y^{(i)}|x^{(i)}, W, b))$$

$$\ell(\theta = \{W, b\}, \mathcal{D}) = -\mathcal{L}(\theta = \{W, b\}, \mathcal{D})$$



Restricted Boltzmann Machines

Energy-Based Model

Energy-Based Model associate a scalar energy to each configuration of the variables of interest.

The energy of the joint configuration is:

$$E(\mathbf{v},\mathbf{h};\theta) = -\sum_{ij} W_{ij} v_i h_j - \sum_i b_i v_i - \sum_j a_j h_j$$

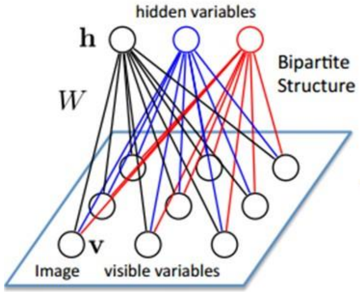
$\theta = \{W, a, b\}$ model parameters.

Probability of the joint configuration is given by the Boltzmann distribution:

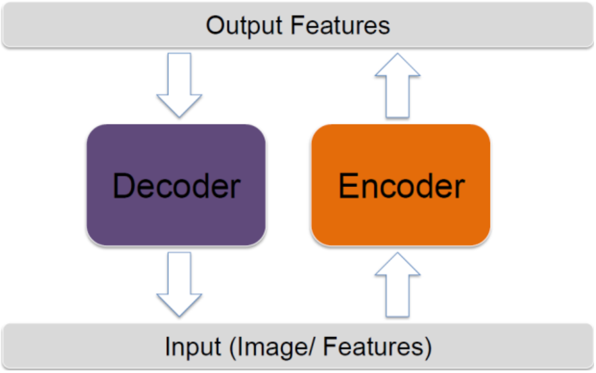
$$P_\theta(\mathbf{v},\mathbf{h}) = \frac{1}{\mathcal{Z}(\theta)} \exp\left(-E(\mathbf{v},\mathbf{h};\theta)\right) = \underbrace{\frac{1}{\mathcal{Z}(\theta)}}_{\text{partition function}} \prod_{ij} \underbrace{e^{W_{ij} v_i h_j}}_{\text{potential functions}} \prod_i e^{b_i v_i} \prod_j e^{a_j h_j}$$

$$\mathcal{Z}(\theta) = \sum_{\mathbf{h},\mathbf{v}} \exp\left(-E(\mathbf{v},\mathbf{h};\theta)\right)$$

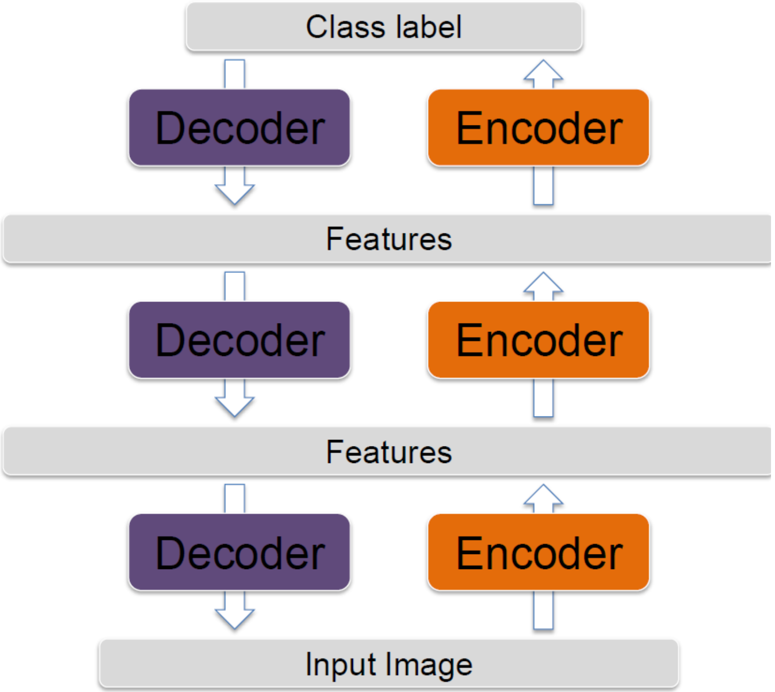
Restricted: No interaction between hidden variables.



Auto-Encoder

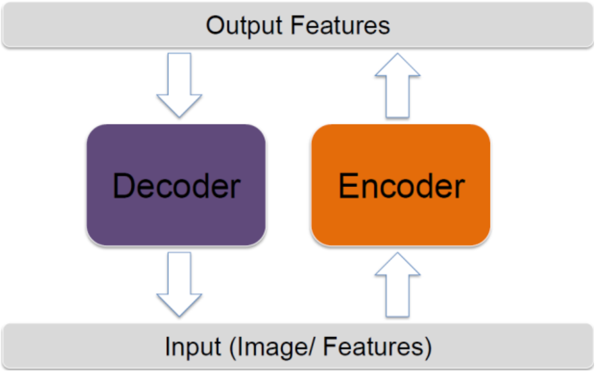


Auto-Encoder

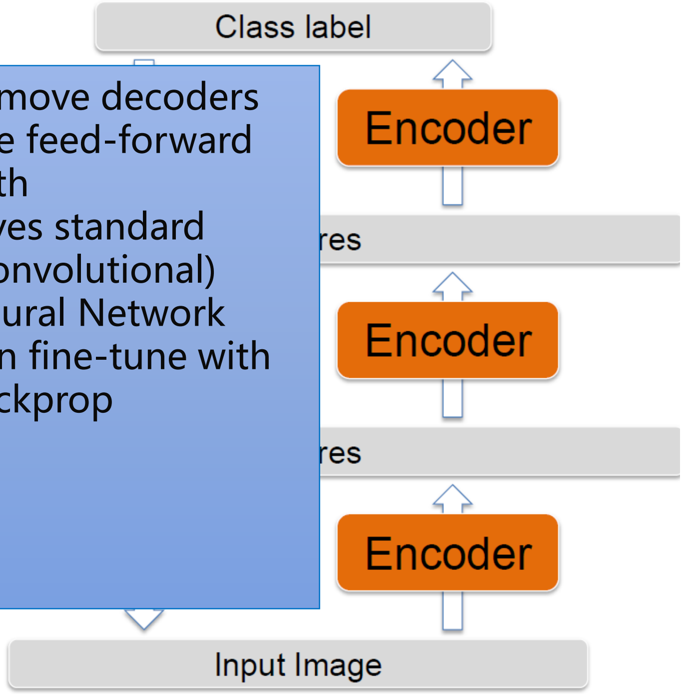


Stacked Auto-Encoders

Auto-Encoder



Auto-Encoder



- Remove decoders
- Use feed-forward path
- Gives standard (Convolutional) Neural Network
- Can fine-tune with backprop

Stacked Auto-Encoders

Convolutional Neural Networks

The Model

Classification is done by projecting an input vector onto a set

$$f(x) = G(b^{(2)} + W^{(2)}(s(b^{(1)} + W^{(1)}x))),$$

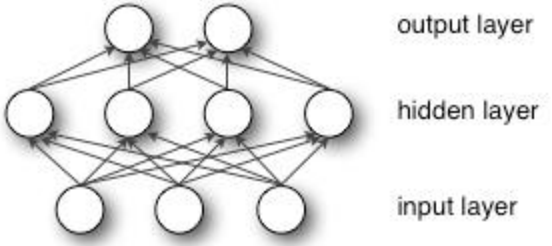
$$y_{pred} = \text{argmax}_i P(Y = i|x, W, b)$$

Loss Function

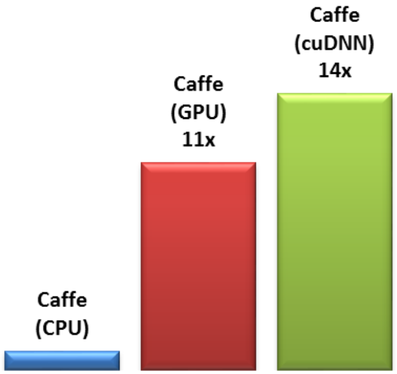
Learning optimal model parameters involves minimizing a loss function.

$$\mathcal{L}(\theta = \{W, b\}, \mathcal{D}) = \sum_{i=0}^{|\mathcal{D}|} \log(P(Y = y^{(i)}|x^{(i)}, W, b))$$

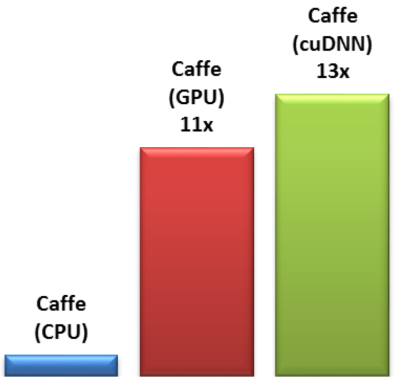
$$\ell(\theta = \{W, b\}, \mathcal{D}) = -\mathcal{L}(\theta = \{W, b\}, \mathcal{D})$$



cuDNN Performance Acceleration



Baseline Caffe compared to Caffe accelerated by cuDNN on K40



Baseline Caffe compared to Caffe accelerated by cuDNN on TitanZ

All comparisons are against a 12-core Intel E5-2679v2 CPU @ 2.4GHz running Caffe with Intel MKL 11.1.3.