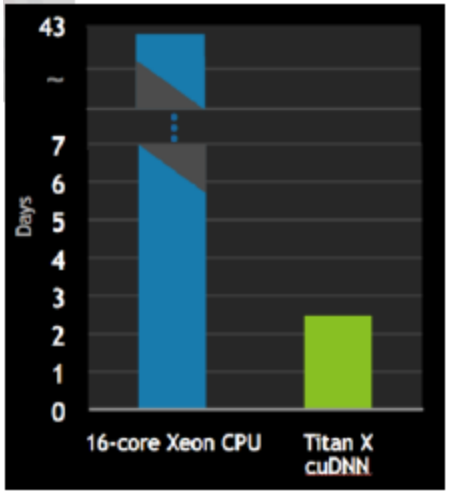
Introduction to CAFFE
A Deep Learning Framework

Tassadaq Hussain
Riphah International University
Barcelona Supercomputing Center
UCERD Pvt Ltd

Hardware Support

Training AlexNet (src: Nvidia)



Which GPU?

Nvidia GPU	Titan X	Tesla K40	Tesla K80
Tflops SP	6.6	4.29	5.6 (total)
Tflops DP	0.2	1.43	1.87 (total)
ECC support	No	Yes	Yes
Memory	12GB	12GB	2 x 12GB
Price (US$)	$1,000	$3,000	$4,200

DL Framework

	Caffe	Theano	Torch
Users	BVLC	Montreal	NYU, FB, Google
Core Language	C++	Python	Lua
Bindings	Python, MATLAB		Python, MATLAB
Pros	Pre-trained models, config files	Symbolic differentiation	
Cons	C++ prototyping, weak RNN support		

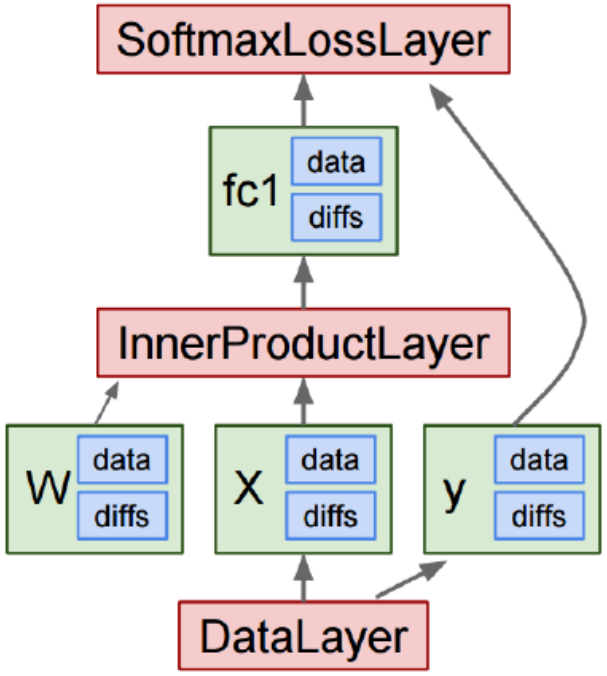
CAFFE: Convolution Architecture For Feature Extraction

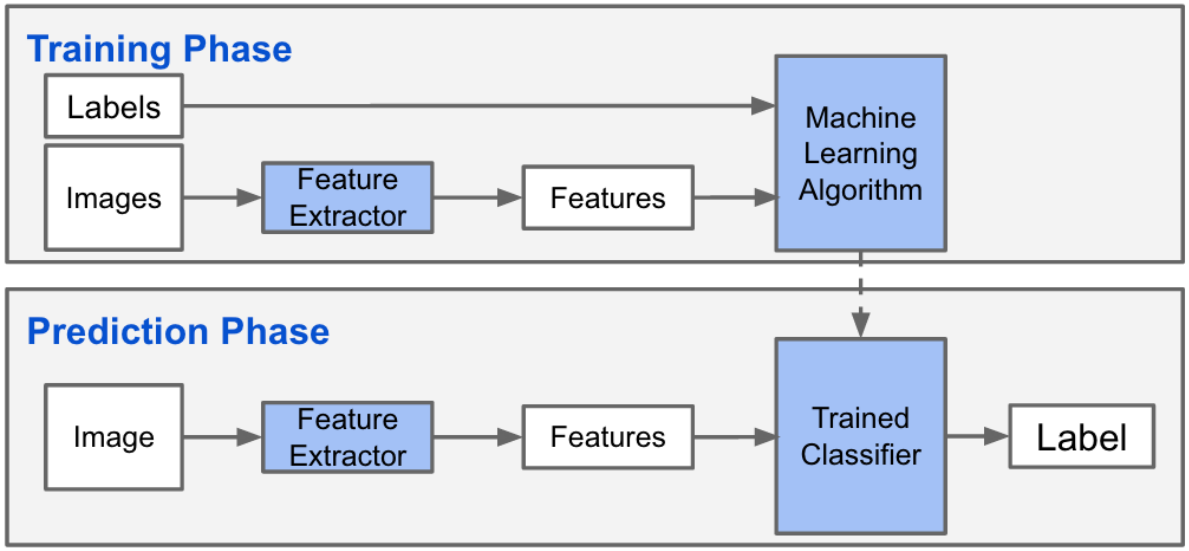
- Open framework, models, and examples for deep learning
- 600+ citations, 100+ contributors, 7,000+ stars, 4,000+ forks
- Focus on vision, but branching out
- Pure C++ / CUDA architecture for deep learning
- Command line, Python, MATLAB interfaces
- Fast, well-tested code
- Tools, reference models, demos, and recipes
- Seamless switch between CPU and GPU

What does Caffe offer

- model definitions
- optimization settings
- pre-trained weights
- so you can start right away.

- **Blob:** Stores data and derivatives
- **Layer:** Transforms bottom blobs to top blobs
- **Net:** Many layers; computes gradients via Forward / Backward
- **Solver:** Uses gradients to update weights

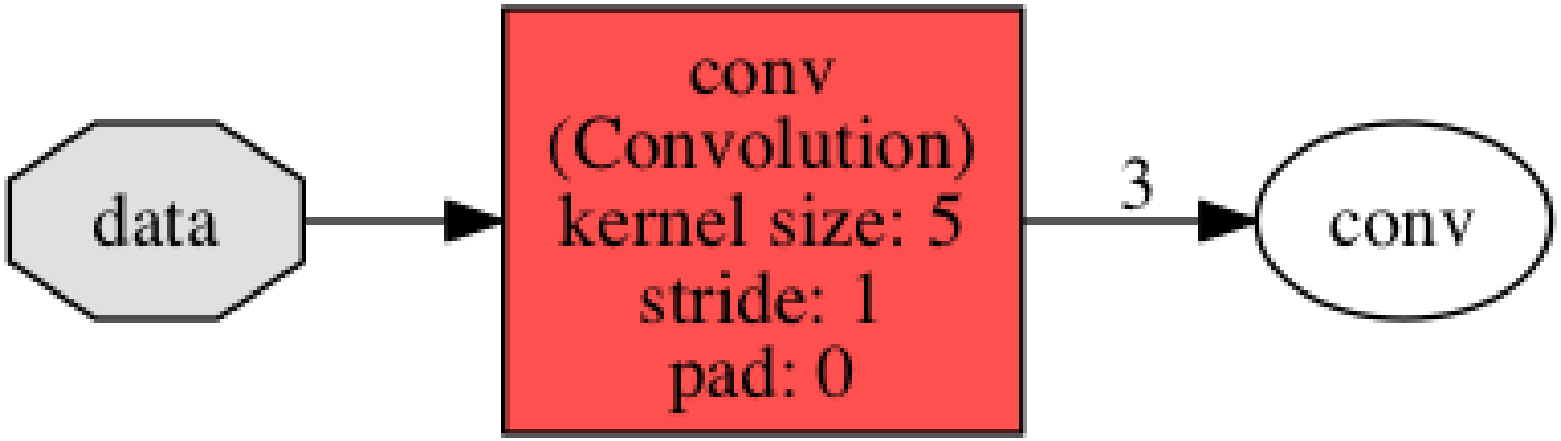




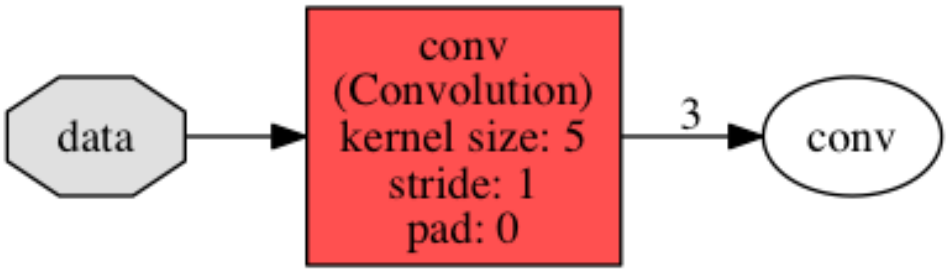
Machine Learning Phases

Traditional Machine Learning Flow

Deep Learning Flow


data
conv
(Convolution)
kernel size: 5
stride: 1
pad: 0
3
conv

- import numpy as np
- import matplotlib.pyplot as plt
- from PIL import Image
- import caffe
- caffe.set_mode_cpu()
- caffe.set_device(0)
- caffe.set_mode_gpu()




```
name: "convolution"
input: "data"
input_dim: 1
input_dim: 1
input_dim: 100
input_dim: 100
layer {
  name: "conv"
  type: "Convolution"
  bottom: "data"
  top: "conv"
  convolution_param {
    num_output: 3
    kernel_size: 5
    stride: 1
    weight_filler {
      type: "gaussian"
      std: 0.01
    }
    bias_filler {
      type: "constant"
      value: 0
    }
  }
}
```

- net = caffe.Net('conv.prototxt', caffe.TEST)