



Caffe Tour

Jan. 15, 2015

Caffe?

- A framework/library of deep learning/machine learning algorithms
 - Written in C++
 - Interface with MATLAB and Python
 - GPU support

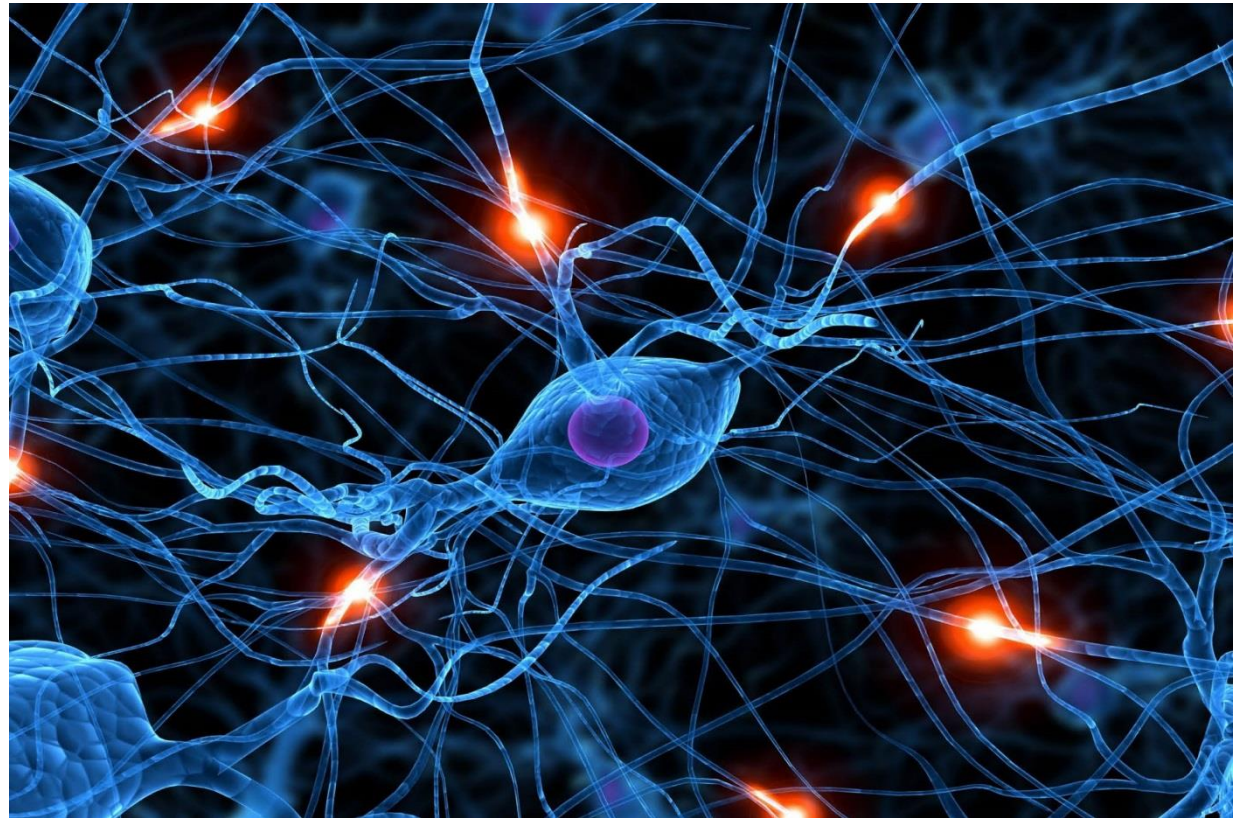
Other libraries

- Theano: Python
- Torch: Lua
- TensorFlow: Google

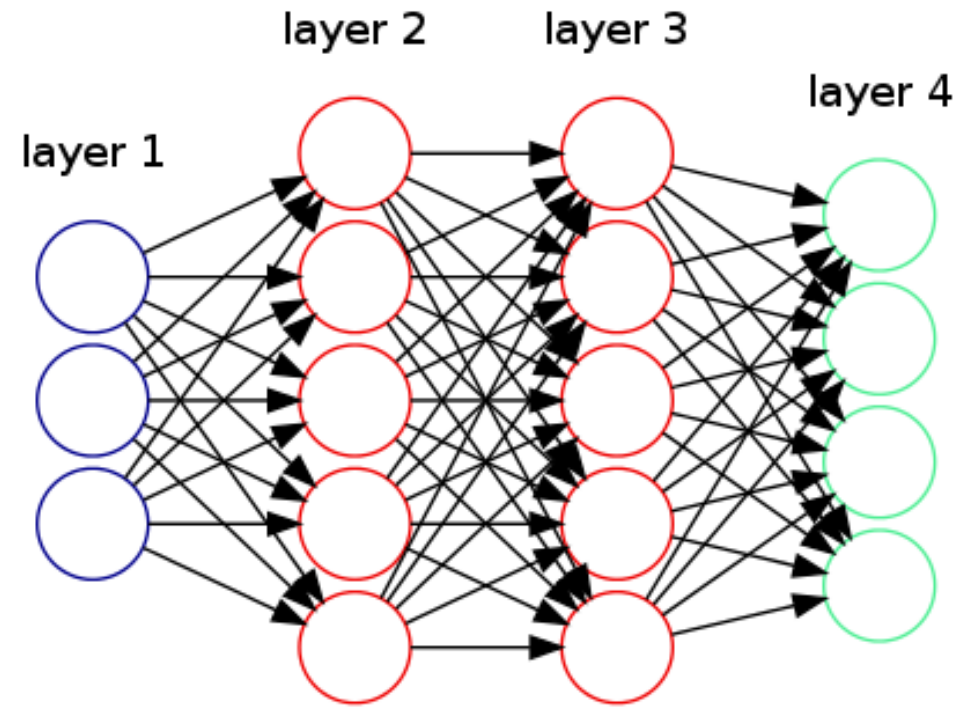


Deep Learning

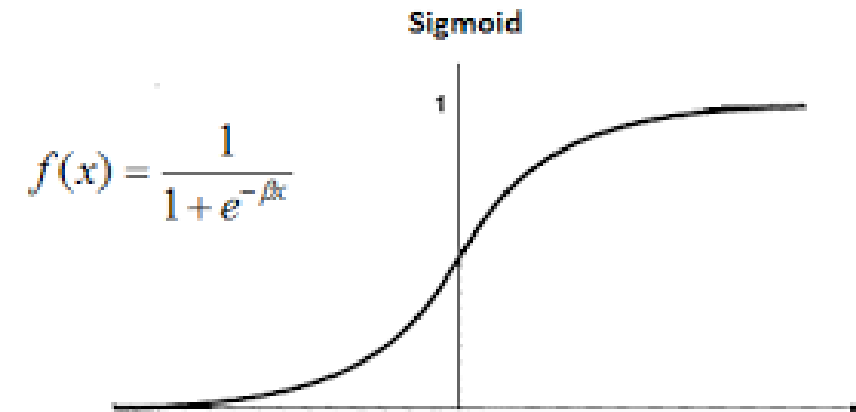
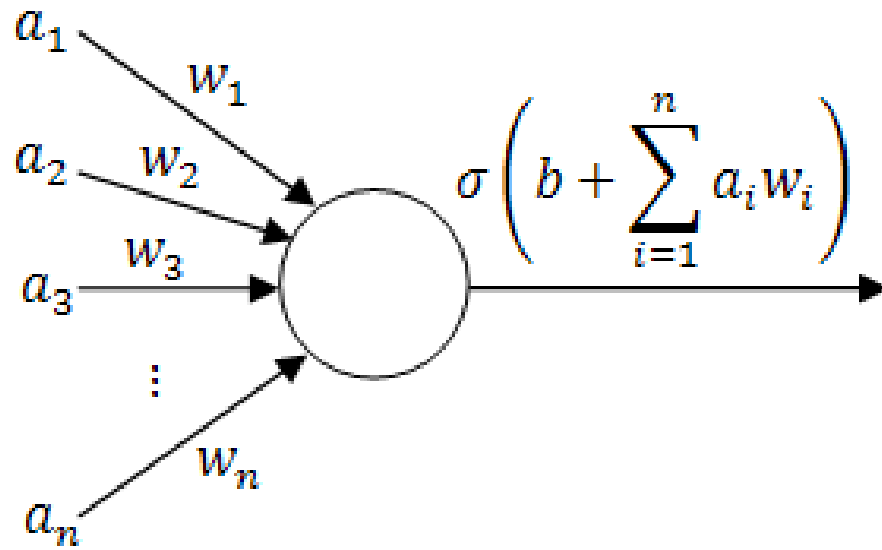
- Algorithms that learn from data
- Neural networks
- Inspired from human brain
- Applications
 - Vision
 - Speech
 - Robotics
 - etc



Neural Networks

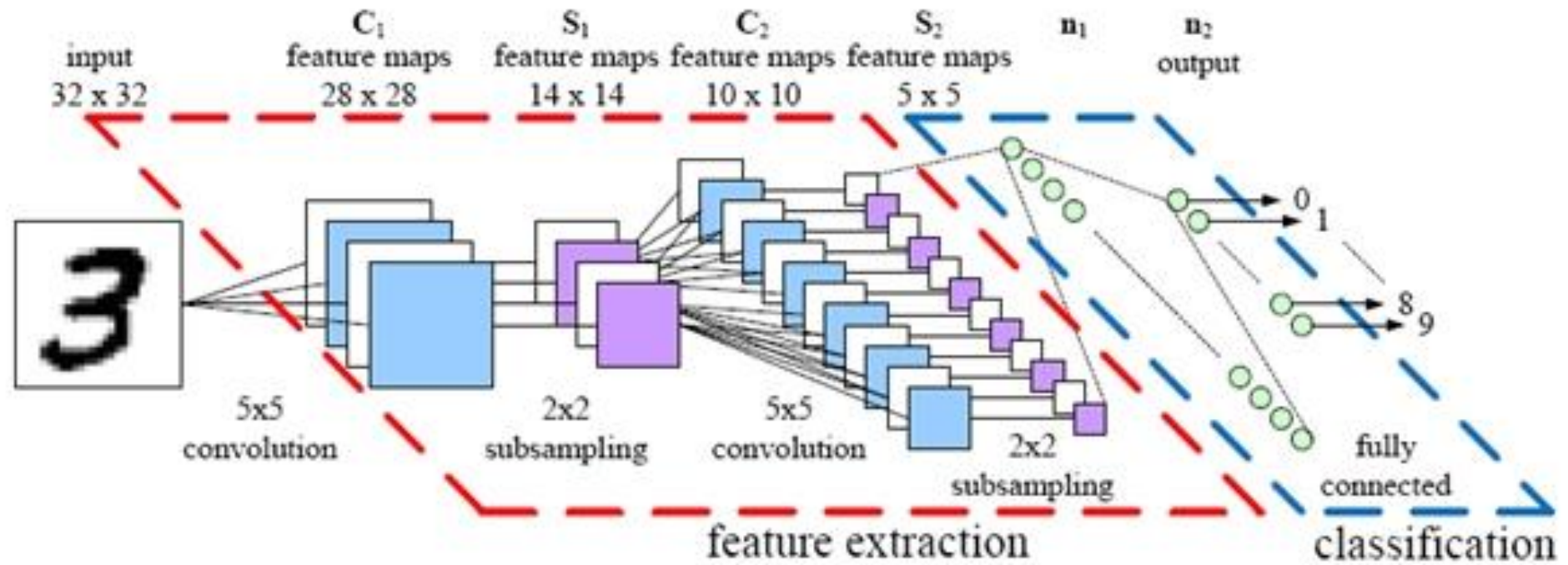


Perceptron



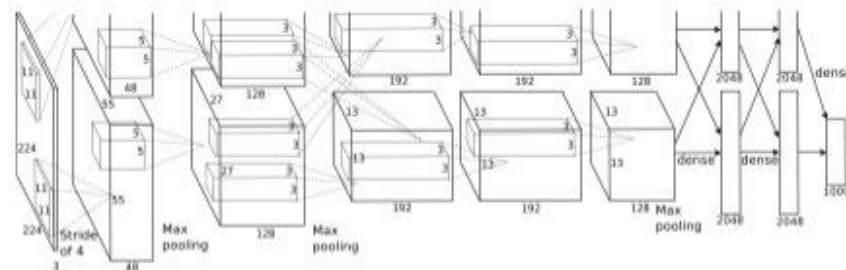
Caffe Process

- Architecture Design



Forward/Backward

Forward:
inference $f_W(x)$



“espresso”
+ loss



$\nabla f_W(x)$ Backward:
learning

Procedure

- Convert the data to Caffe-format
 - Imdb, leveldb, hdf5 / .mat, list of images, etc.
- Define the Net
- Configure the Solver
- `caffe train -solver solver.prototxt -gpu 0`

Net: *.prototext* file

```
layer {  
  name: "conv1"  
  type: "Convolution"  
  param { lr_mult: 1 }  
  param { lr_mult: 2 }  
  convolution_param {  
    num_output: 20  
    kernel_size: 5  
    stride: 1  
    weight_filler {  
      type: "xavier"  
    }  
    bias_filler {  
      type: "constant"  
    }  
  }  
  bottom: "data"  
  top: "conv1"  
}
```

Solver

26 lines (25 sloc) | 790 Bytes

Raw

Blame

History



```
1 # The train/test net protocol buffer definition
2 net: "examples/mnist/lenet_train_test.prototxt"
3 # test_iter specifies how many forward passes the test should carry out.
4 # In the case of MNIST, we have test batch size 100 and 100 test iterations,
5 # covering the full 10,000 testing images.
6 test_iter: 100
7 # Carry out testing every 500 training iterations.
8 test_interval: 500
9 # The base learning rate, momentum and the weight decay of the network.
10 base_lr: 0.01
11 momentum: 0.9
12 weight_decay: 0.0005
13 # The learning rate policy
14 lr_policy: "inv"
15 gamma: 0.0001
16 power: 0.75
17 # Display every 100 iterations
18 display: 100
19 # The maximum number of iterations
20 max_iter: 10000
21 # snapshot intermediate results
22 snapshot: 5000
23 snapshot_prefix: "examples/mnist/lenet"
24 # solver mode: CPU or GPU
25 solver_mode: GPU
```

Pre-Trained Nets

- Import previous solver state
- Continue training
- Fine-tuning

Digit Recognition

- Training: 60,000 images
- Test: 10,000 images