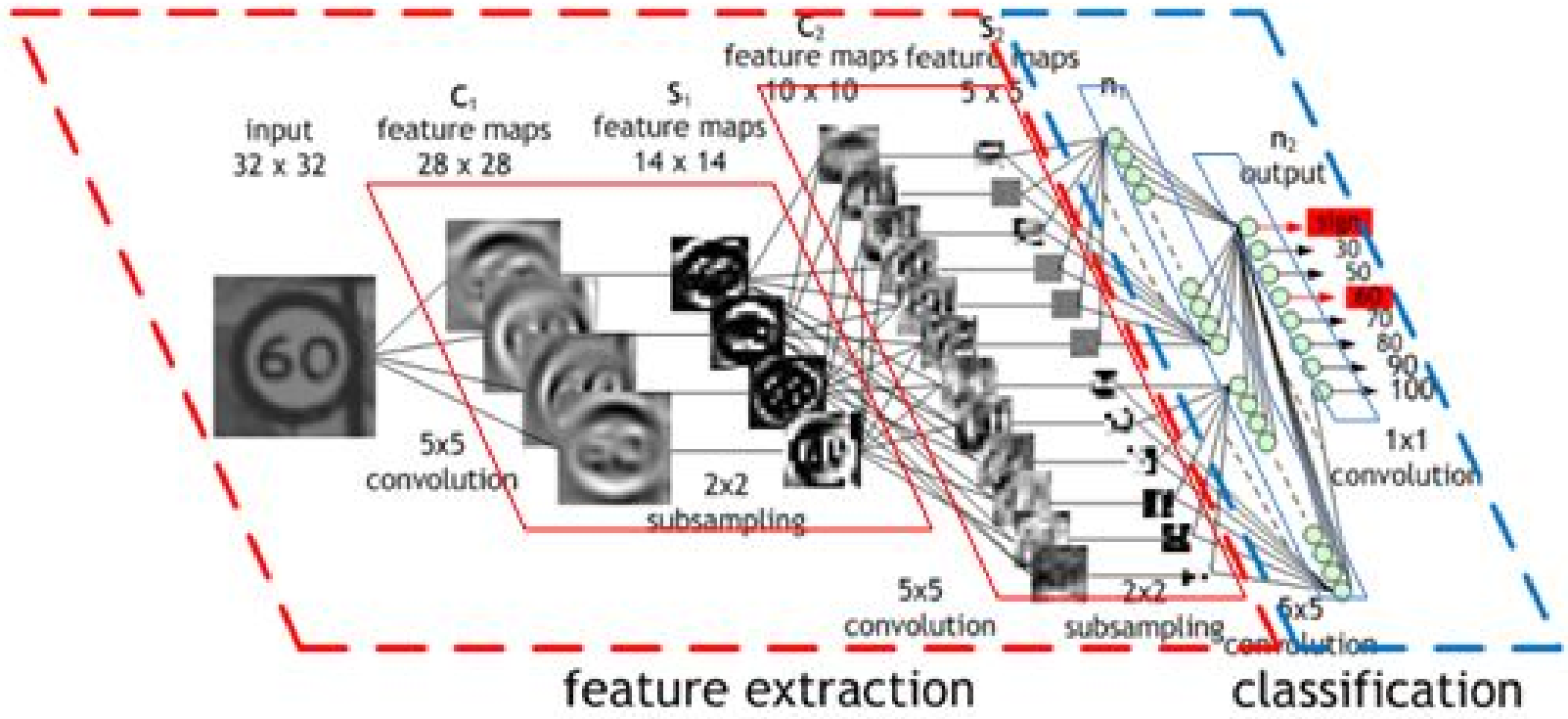
GPU-Accelerated Deep Convolutional Neural Network for Object Recognition

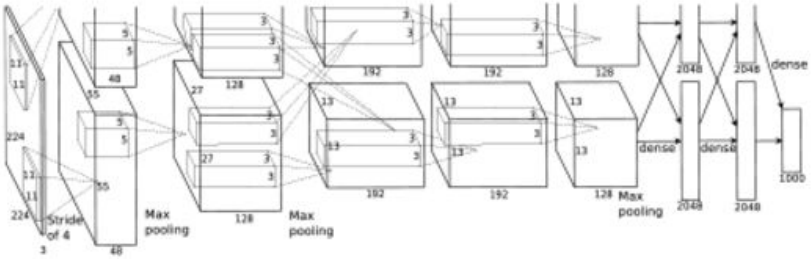
Guan Sun

Convolutional Neural Network



How does it work?

Forward:
inference $f_W(x)$



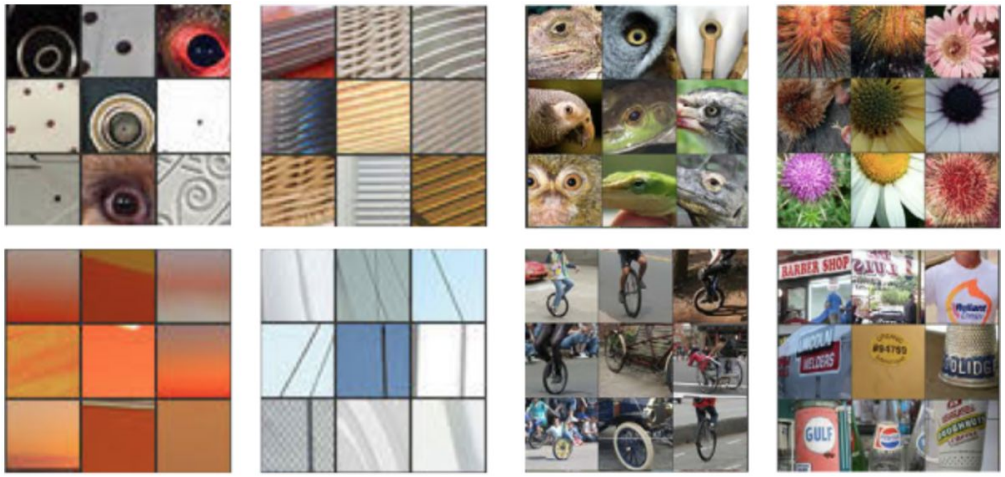
"espresso"
+ loss

$\nabla f_W(x)$ Backward:
learning

Why Deep? Why GPU?

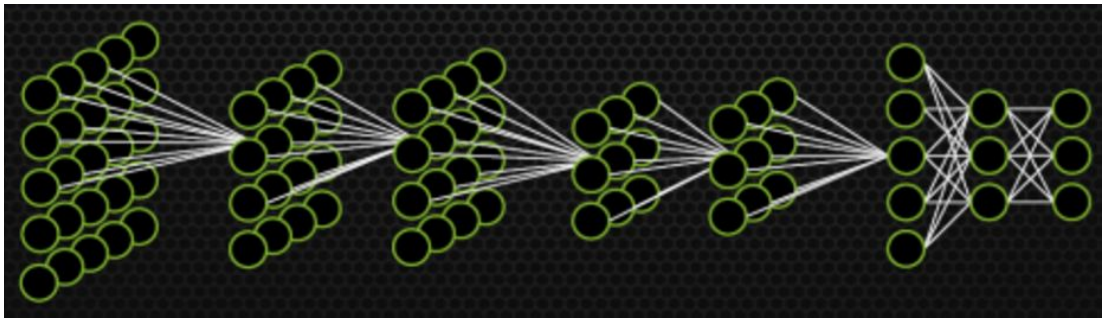
Depth make a difference

Low level



High level

Neurons are highly parallel by nature



Caffe

A open source framework for deep learning.

A Caffe Layer

```
name: "conv1"
type: CONVOLUTION
bottom: "data"
top: "conv1"
convolution_param {
    num_output: 20
    kernel_size: 5
    stride: 1
    weight_filler {
        type: "xavier"
    }
}
```

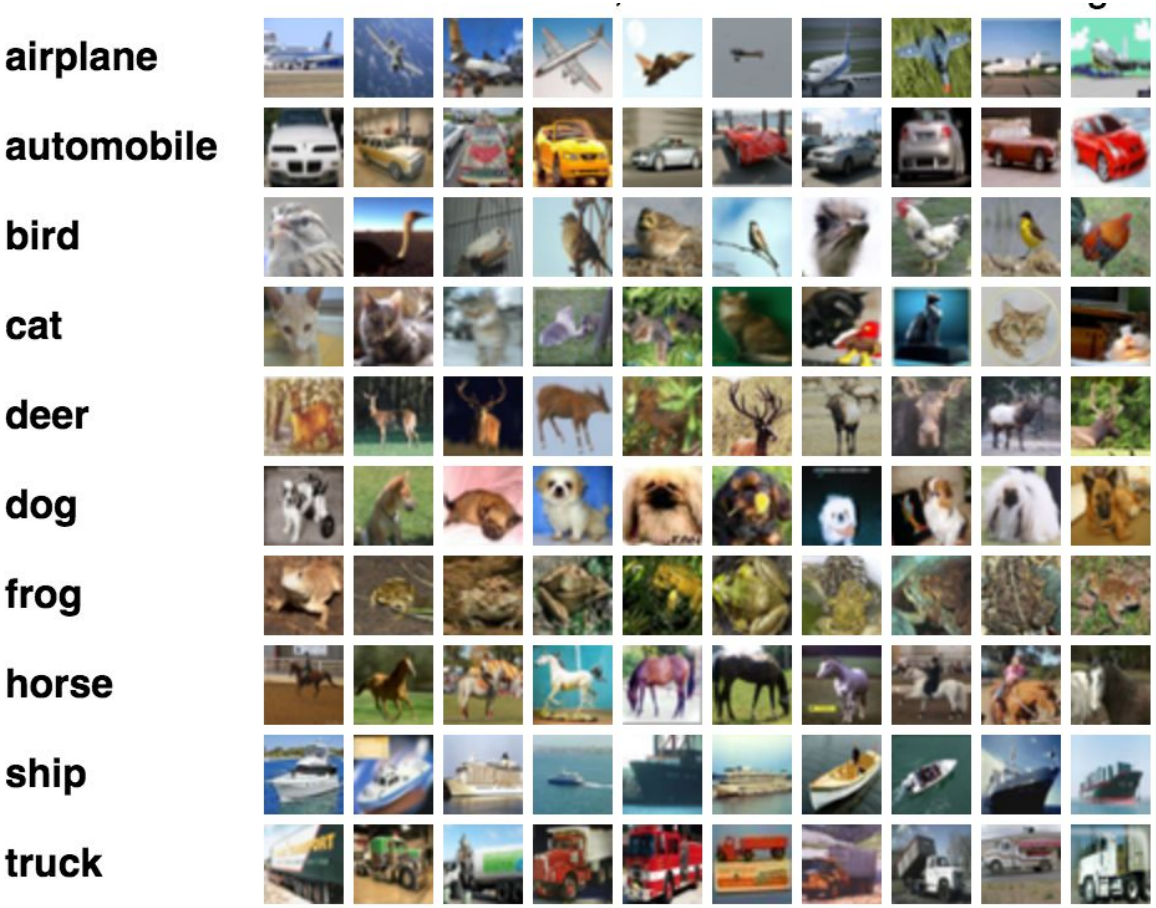
A Caffe Network

- A network is a set of layers connected as a DAG:

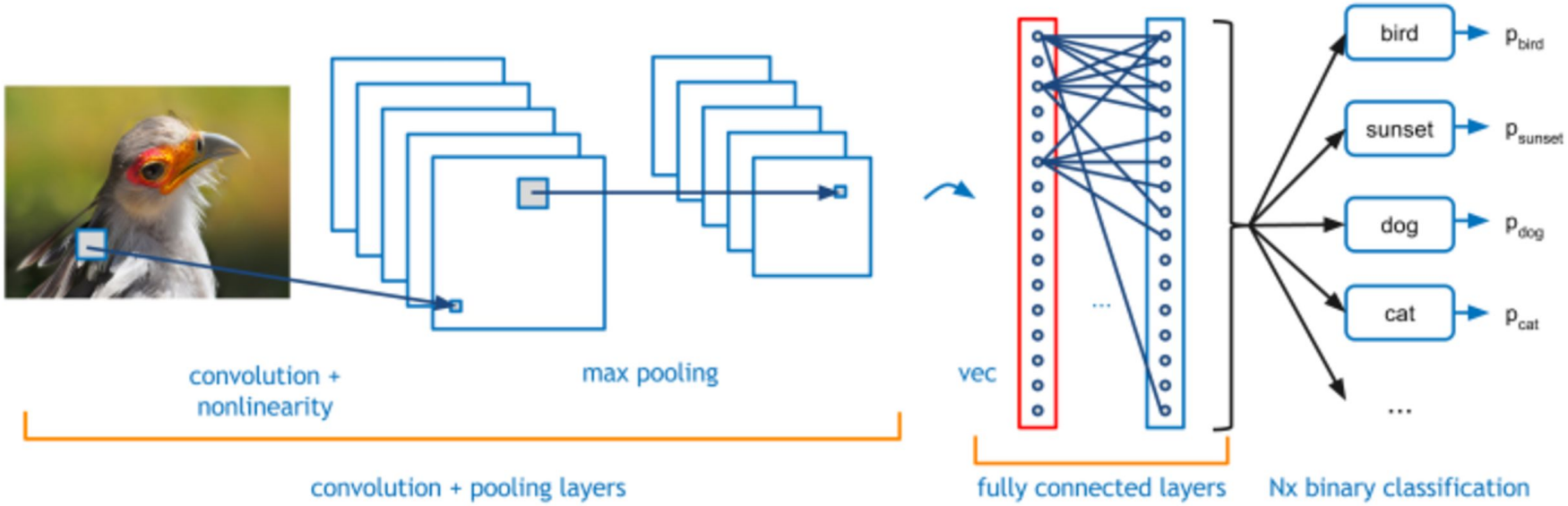
```
name: "dummy-net"
layers { name: "data" …}
layers { name: "conv" …}
layers { name: "pool" …}
    … more layers …
layers { name: "loss" …}
```

Object Recognition

The CIFAR-10 dataset



Object Recognition



Demo

```
---------- Prediction for examples/images/cat.jpg
0.3134 - "n02123045 tabby, tabby cat"
0.2380 - "n02123159 tiger cat"
0.1235 - "n02124075 Egyptian cat"
0.1003 - "n02119022 red fox, Vulpes vulpes"
0.0715 - "n02127052 lynx, catamount"
```

```
---------- Prediction for examples/images/bottle.jpg
0.9001 - "n04591713 wine bottle"
0.0252 - "n02823428 beer bottle"
0.0240 - "n03983396 pop bottle, soda bottle"
0.0070 - "n04557648 water bottle"
0.0070 - "n03916031 perfume, essence"
```

Roadmap

Milestone 1 (11/23):

Implement computation for Convolutional layer and Pooling layer

Milestone 2 (11/30):

Implement computation for LeRU layer, Inner product layer and Softmax layer

Milestone 3 (12/07):

Performance analysis vs other CUDA implementation and CPU implementation

Final results (12/11):

CUDA implementation of CNN under Caffe framework

A trained deep CNN model for object recognition

Detailed performance analysis

Thanks!