Master of Technology

Unit 2/6: Computational Intelligence I

Advanced Topics

Dr. Zhu Fangming
Institute of Systems Science
National University of Singapore

Objectives

- To introduce the basic concepts of rule extraction/ generation from neural networks
- To introduce neural and fuzzy hybrid architectures
- To introduce Convolutional Neural Network (CNN)

Outline

- Rule Extraction from NNs
- Neural Networks and Fuzzy Systems
- Convolutional Neural Network (CNN)

Rule Generation — A Brief Overview

- **Rule generation has become an important topic in both machine learning and data mining research**
  - » **Machine learning concerns the construction of computer programs that learn knowledge or skill and present it in terms of 'if-then' rules**
  - » **Data mining is the process of the discovery of, non-obvious and potentially useful knowledge/ patterns hidden in the data**
  - » **The preponderance of neural networks in several fields and the inherent 'black-box' nature of some of them made researchers to come out with methods and techniques to extract rules from NN. This is to depict the knowledge learned by an NN in a human comprehensible manner**
    - ♦ **Has many applications in medicine and other mission-critical applications**
  - » **The emergence of a variety of fuzzy neural networks and neuro-fuzzy systems paved the way for a sophisticated expert systems with enormous features based on the principles of soft computing paradigm.**

Rule Learning Methods

- **Rule learning methods**
  - » **Symbolic heuristic search**
    - ♦ **E.g. hill-climbing search**
  - » **Decision trees**
    - ♦ **The decision tree is constructed by sequentially selecting attributes and recursively partitioning them. It is translated into a set of rules. CART, C5.0 are examples.**
  - » **Neural networks**
    - ♦ **Extract rules from a trained neural network**
  - » **Clustering** (some NN can be used for clustering)
    - ♦ **Each cluster shows a potential rule**

Rule Extraction from Neural Networks

- **A major weakness of the neural network approach to artificial intelligence is that the knowledge learned by a neural network is difficult to interpret.**

- **It is mandatory for neural networks to have explanation capability besides functionality in safety critical applications such airlines traffic control, operation of power stations and medical diagnosis and surgery.**

- **The rules extracted from an NN can also be used to build a knowledge-based neural network which outperforms the original network. According to Towell and Shavlik (1993), it is because the rule-extraction procedure reduces the over-fitting of the trained examples.**

- **The rule extraction algorithm will depend on the kind of rule to be extracted.**

- **Rule search program extracts valid rules (may be certain or uncertain) with or without a degree of certainty.**

Rule Extraction from Neural Networks

- **The form of a rule generated from NN:**

  If A^+_1, …, A^+_i, …, $\neg A^-_1$, …, $\neg A^-_j$, …, Then C (or ¬C)

  where

  A^+_i is a positive antecedent (an attribute in the positive form)

  A^-_j is a negative antecedent (an attribute in the negative form)

  C is the concept, and ¬C means "not C"

- **A rule is a *confirming* (*disconfirming*) rule if the action is C (¬C)**

- **The connection between attributes is conjunction (AND)**

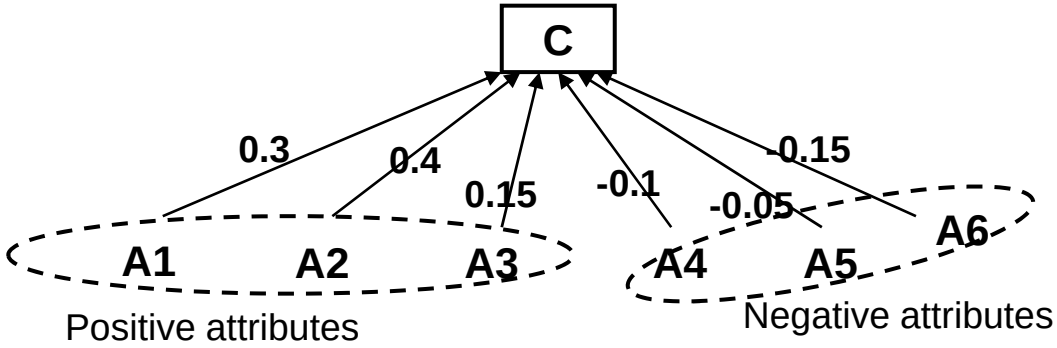
- **Multiple rules with the same conclusion represent disjunction (OR).**

Rule Extraction Algorithm

- **A general rule extraction algorithm KT (Fu, 1991)**
  - » **KT generates rules from an NN on a layer-by-layer basis. In each layer, it explores up to *k* (*k* is predefined) rules. The largest-possible number of rules relating the input attributes and the output concepts is k^d (d is the number of layers).**
  - » **Rule size is defined to be the total number of attributes mentioned in the rule's premise.**
  - » **KT generates rules for each concept corresponding to a hidden or output unit (called a concept node). For each concept, KT explores combinations of relevant attributes (positive attributes and negative attributes) systematically by conducting a tree search.**
  - » **Each node in the tree represents a combination of attributes.**
  - » **A node at the *i*-th level generates its child nodes at the level *i+1*-th level by adding an additional, available attribute in all possible ways**

A Simple Example of Rule Extraction

- **Rule extraction by KT algorithm**
  - » **given a simple neural network unit**
  - » **threshold of the unit $\alpha = 0.5$**

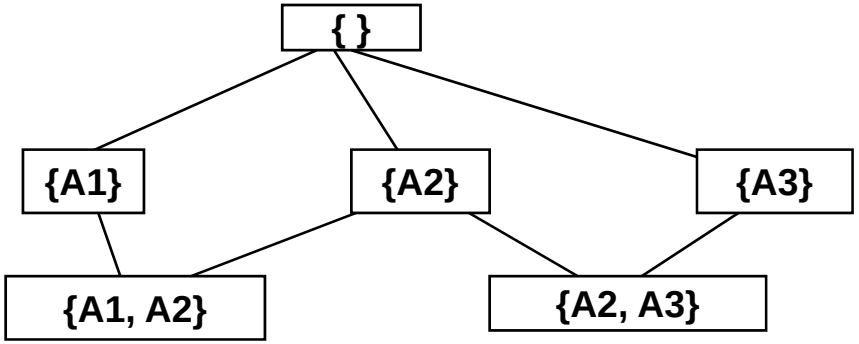


Step-1:

- » **a set of all positive attributes S_P = {A1, A2, A3}**
- » **a set of all negative attributes S_N = {A4, A5, A6}**

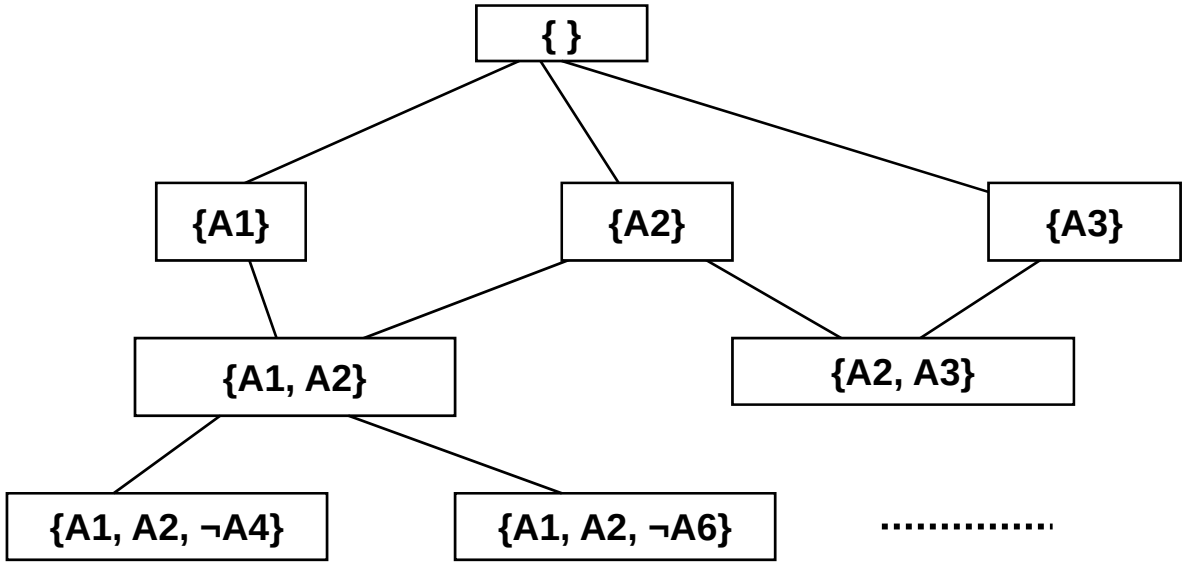
A Simple Example of Rule Extraction (cont.)

- **Step-2:**
  - » search a set of positive attributes $S_p \subseteq S_P$ whose summed weights exceed the threshold
  - » E.g.: {A1, A2} with summed weights 0.3 + 0.4 = 0.7 > 0.5



- **Step-3:**
  - » for each positive attribute set, add one negative attribute A_n^- satisfying the condition:
    - ♦ the summed weights of the positive attribute set plus $S_N - \{A_n^-\}$ exceed the threshold on the unit
  - » E.g.: for {A1, A2, ¬A4}: 0.3 + 0.4 - 0.05 - 0.15 = 0.5
  - » E.g.: for {A1, A2, ¬A6}: 0.3 + 0.4 - 0.05 - 0.1 = 0.55

A Simple Example of Rule Extraction (cont.)



- **Step-4:**
  - » **form rules:**

    If A1 and A2 and not A4, Then C

    If A1 and A2 and not A6, Then C

Rule Extraction from Neural Networks

- **Three categories of rule extraction techniques:**

Decompositional techniques

» **The focus is on extracting rules at the level of hidden and output nodes within the trained NN**

» **Thus, the trained NN is *transparent***

» **Here, the computed output from hidden and output nodes must be mapped into a binary outcome that corresponds to a rule consequent**

Rule Extraction from Neural Networks

» **Each hidden or output node is interpreted as a step function or a Boolean rule that reduces the rule extraction problem to one of determining the situations where the rule is 'true'**

» **A set of incoming links whose summed weights guarantee the unit's bias is exceeded regardless of the activation value of other incoming links**

» **The rule extracted at the individual nodes are then aggregated to form a composite rule base for the NN.**

Rule Extraction from Neural Networks

Pedagogical techniques

» **These methods treat the trained NN as a black box**

» **The view of the trained NN is *opaque***

» **Here rule extraction is viewed as a learning task where the target concept is the function computed by the network and the input features are simply networks' input features**

» **Hence these methods aim to extract rules that map inputs directly into outputs**

» **These are typically used in conjunction with a symbolic learning algorithm and the trained NN is used to generate examples for the learning algorithm**

Rule Extraction from Neural Networks

Eclectic techniques

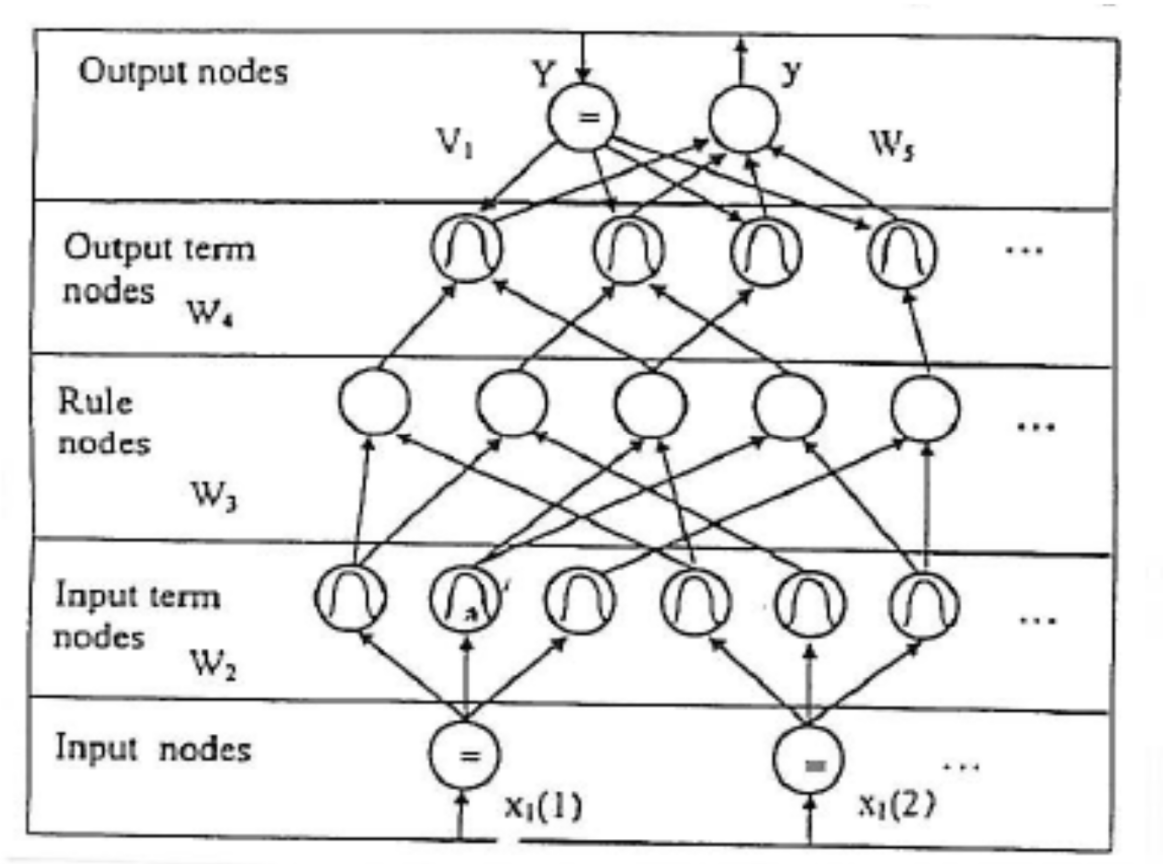
» They combine both decompositional and pedagogical techniques

» They use the knowledge about the internal architecture and /or weight vectors in the trained NN to complement a symbolic learning algorithm

Neural Networks and Fuzzy Systems

- **Complementary technologies**
  - » **Neural Networks (NN)**
    - ♦ **Essentially computational algorithms**
    - ♦ **Extract information from systems to be learned or controlled**
    - ♦ **Good for simulating human senses**
  - » **Fuzzy Systems (FS)**
    - ♦ **Use verbal and linguistic information from experts**
    - ♦ **Provide a structured framework that utilizes and exploits these capabilities of NNs**
    - ♦ **Good for mimicking human thinking, reasoning**
- **Advantages**
  - » **NN — learning and optimization abilities, connectionist structures**
  - » **FS — Humanlike 'IF-THEN-rules' thinking, ease of incorporating expert knowledge**
- **How can we bring them closer**
  - » **NN — improve their transparency**
  - » **FS — self-adapt**

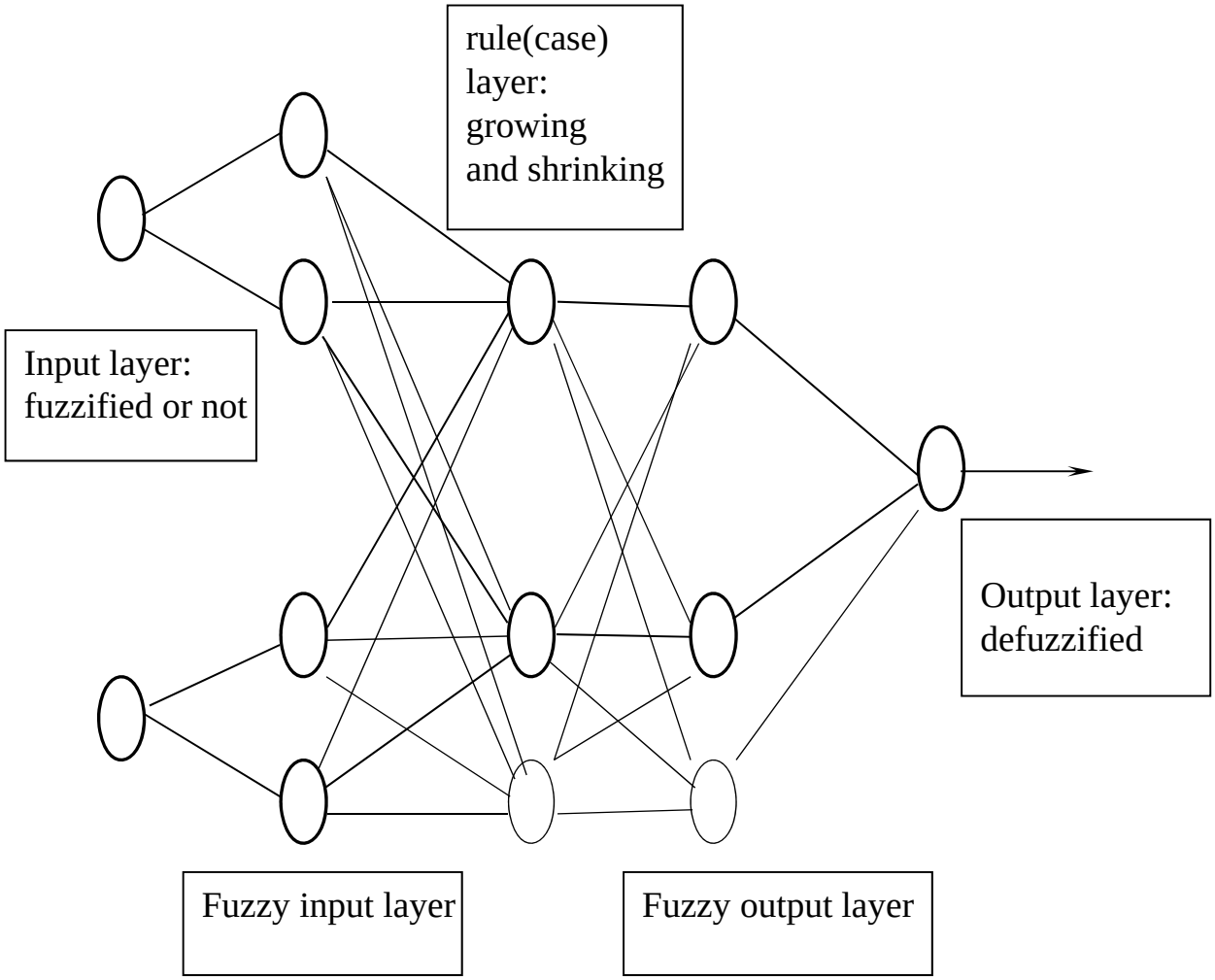
Fuzzy Neural Network - Example



Evolving Connectionist Systems (ECOS)

- These are connectionist systems that combine other paradigms and perform clustering, rule based inference/ classification/ regression and data mining and they function in real time and can handle continuous input streams of data
- Local learning and one-pass learning are the outstanding hallmarks of all such systems
  - EFuNN (Evolving Fuzzy Neural Networks)
  - DENFIS (Dynamic Evolving Neuro-Fuzzy Inference System)
  - ESOM (Evolving SOM)
  - ECM (Evolving Clustering Methods)
- www.theneucom.com

EFuNN Architecture



Convolutional Neural Network (CNN)

- **Convolutional Neural Networks (CNN) are essentially neural networks that use convolution in place of general matrix multiplication in at least one of their layers.**
- **CNN are based on the following principles:**
  - » **Local receptive field**
  - » **Shared weights**
  - » **Pooling (or down sampling)**
- **Applications of CNN**
  - » **Image Recognition**
  - » **Video Analysis**
  - » **Natural Language Processing**
  - » **Computer Board Games (Go, Chess, Shogi)**
  - » **……**

Convolutional Neural Network (CNN)

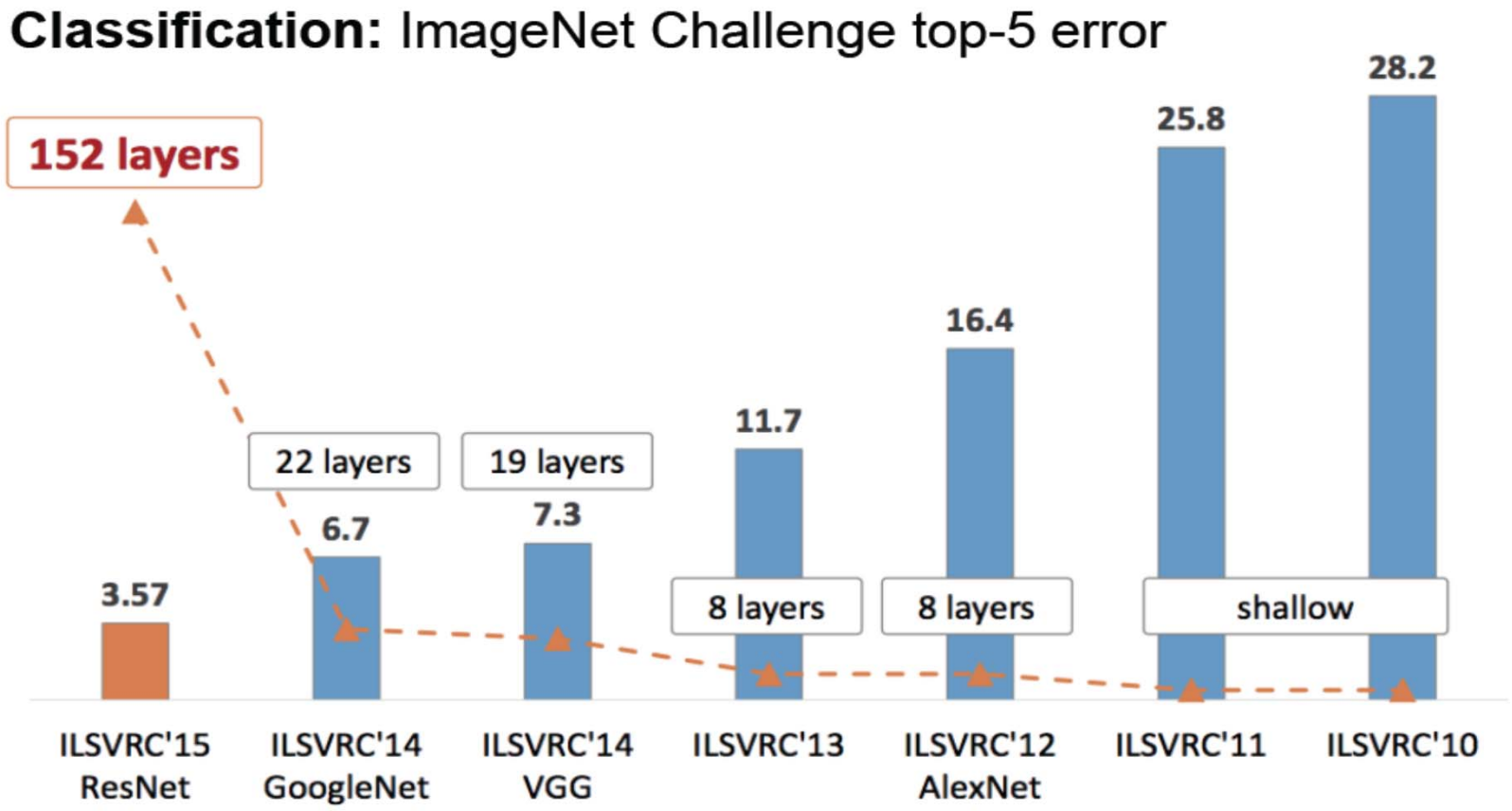
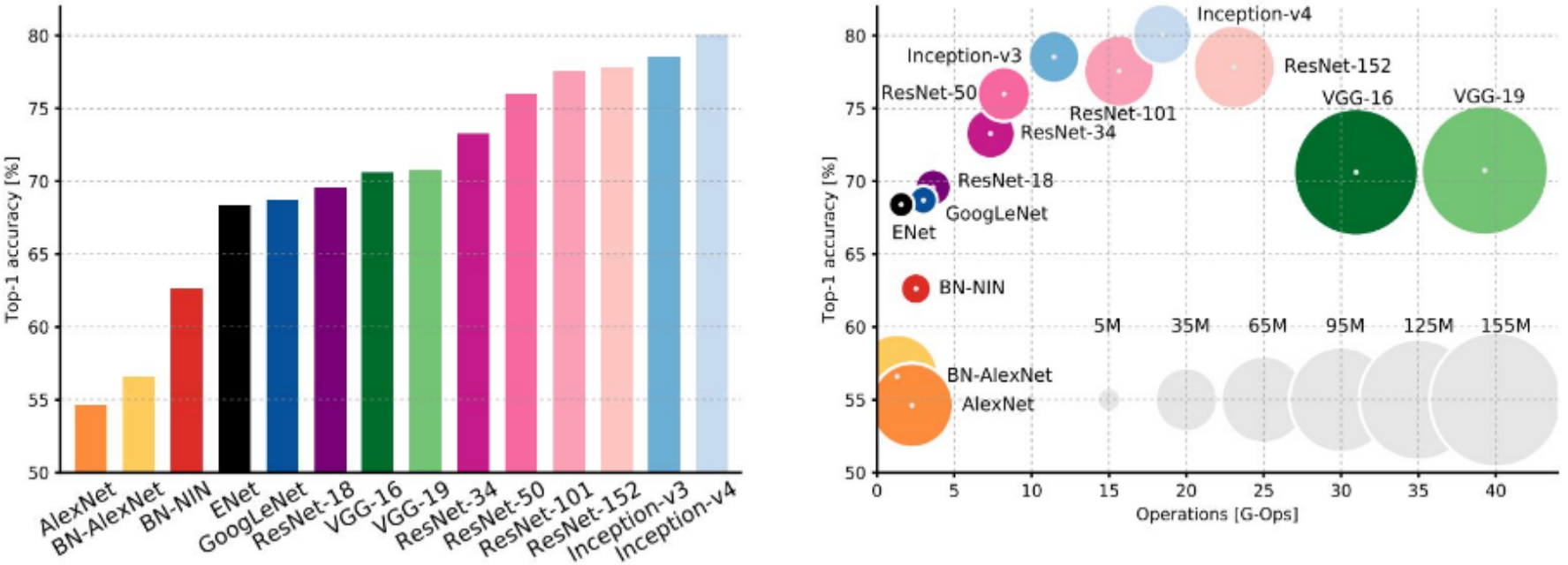


Figure source: Kaiming He

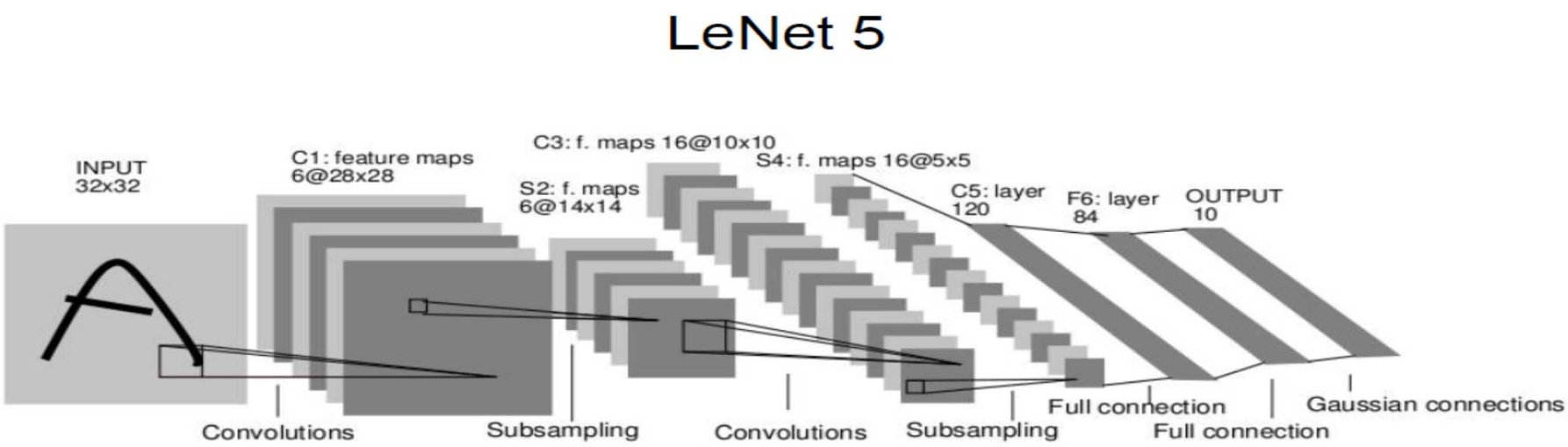
Convolutional Neural Network (CNN)



An Analysis of Deep Neural Network Models for Practical Applications, 2017.

https://medium.com/@siddharthdas_32104

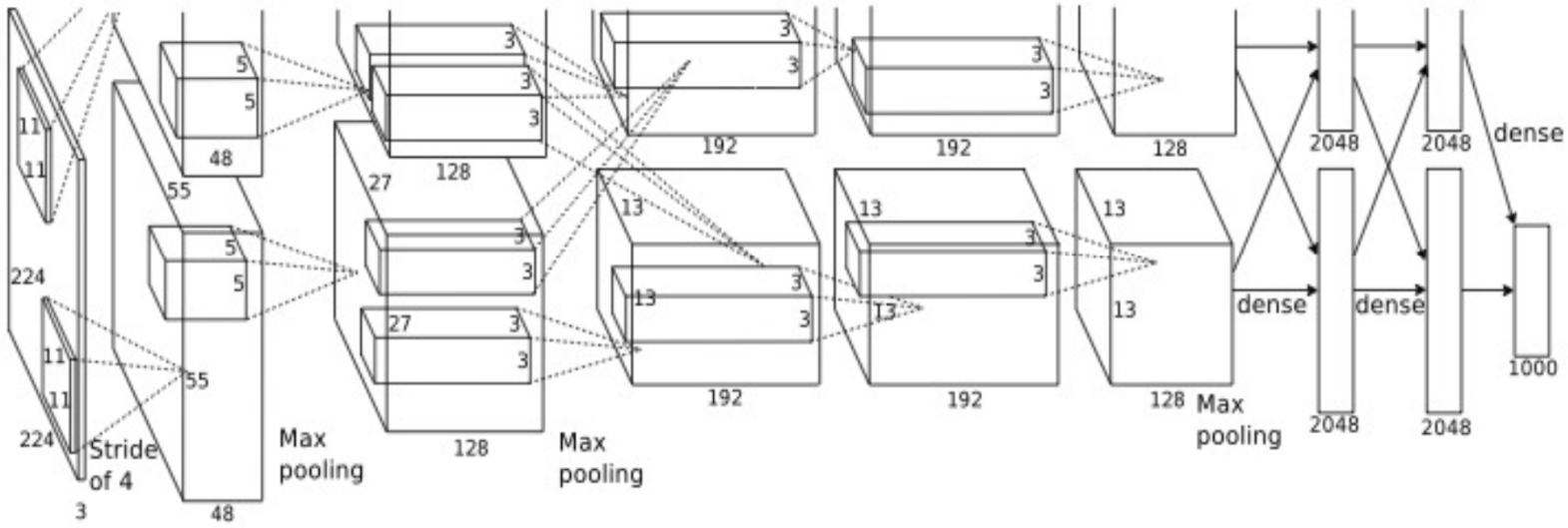
Convolutional Neural Network (CNN) – LeNet -5



LeNet-5, LeCun et al. 1998

- Average pooling
- Sigmoid or tanh nonlinearity
- Fully connected layers at the end
- Trained on MNIST digit dataset with 60K training examples

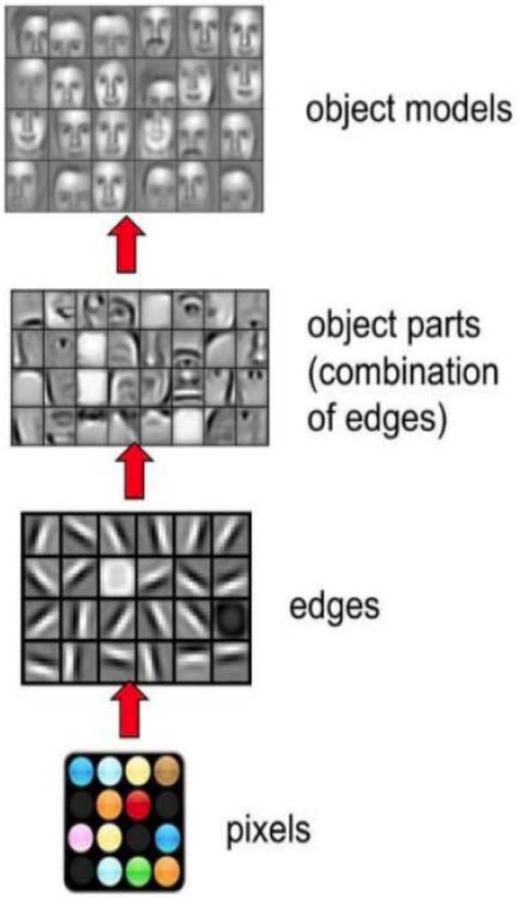
Convolutional Neural Network (CNN) - AlexNet

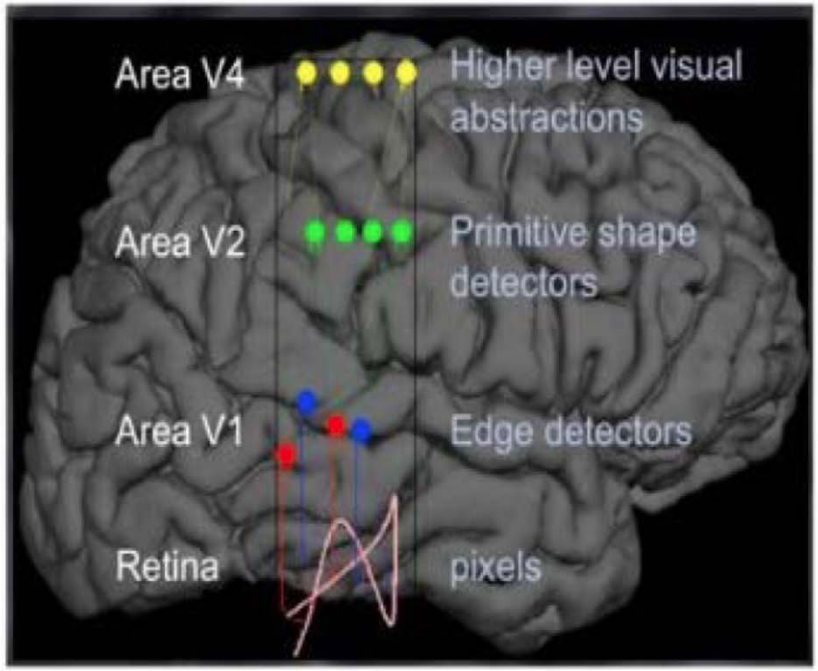


AlexNet, Krizhevsky et al. 2012

- Similar framework to LeNet but:
  - Max pooling, ReLU nonlinearity
  - More data and bigger model (7 hidden layers, 650K units, 60M params)
  - GPU implementation (50x speedup over CPU)
    - Trained on two GPUs for a week
  - Dropout regularization

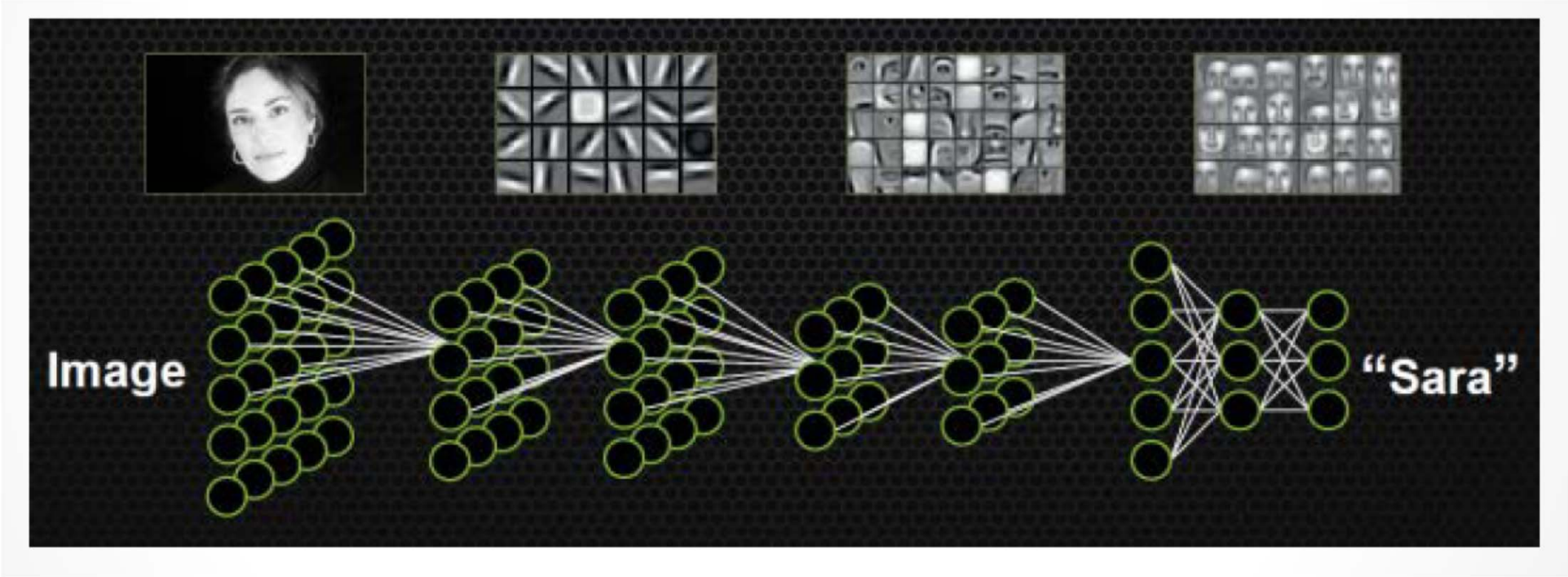
Convolutional Neural Network (CNN)





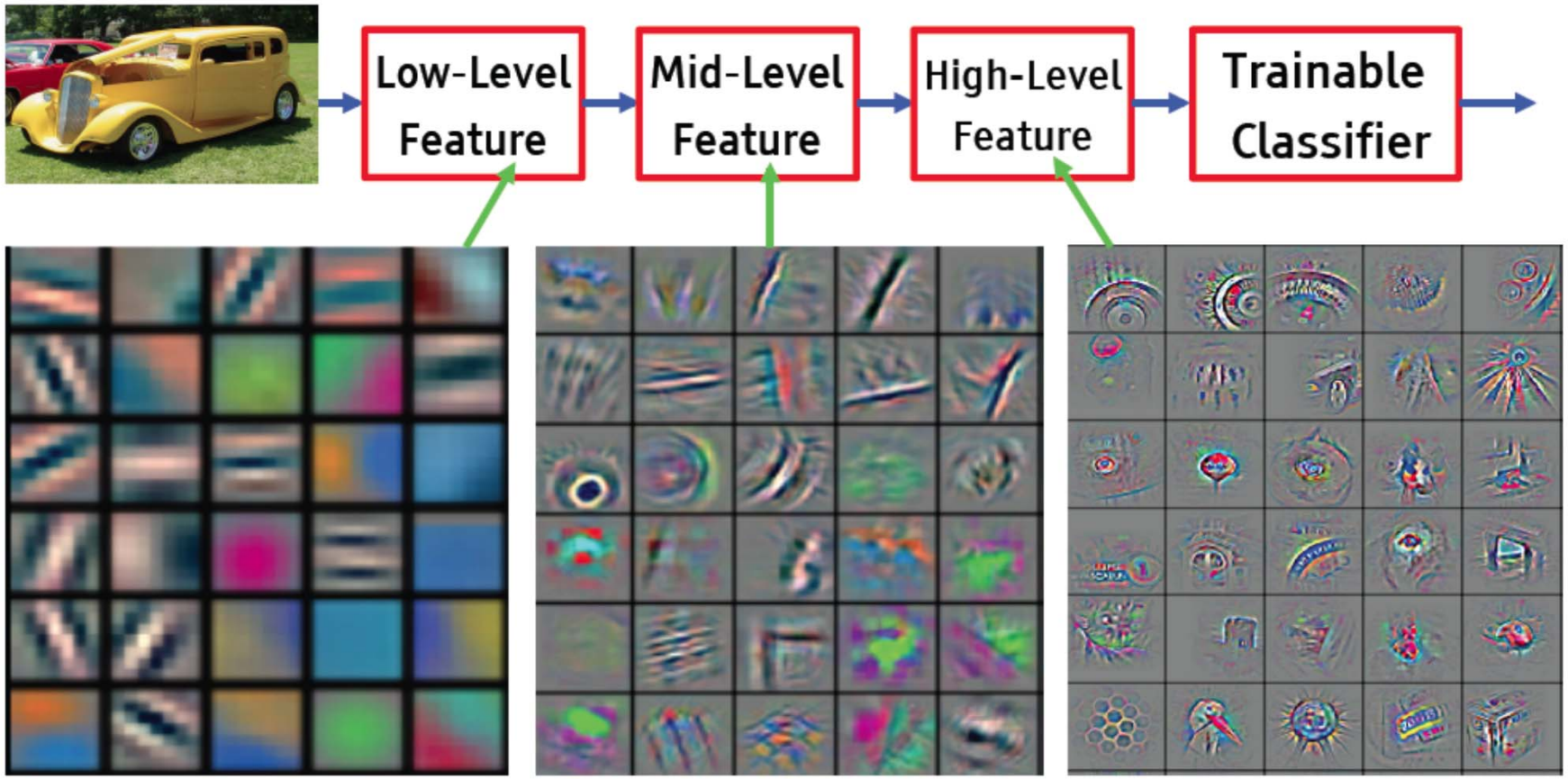
Andrew Ng: Deep Learning, Self-Taught Learning and Unsupervised Feature Learning

Convolutional Neural Network (CNN)



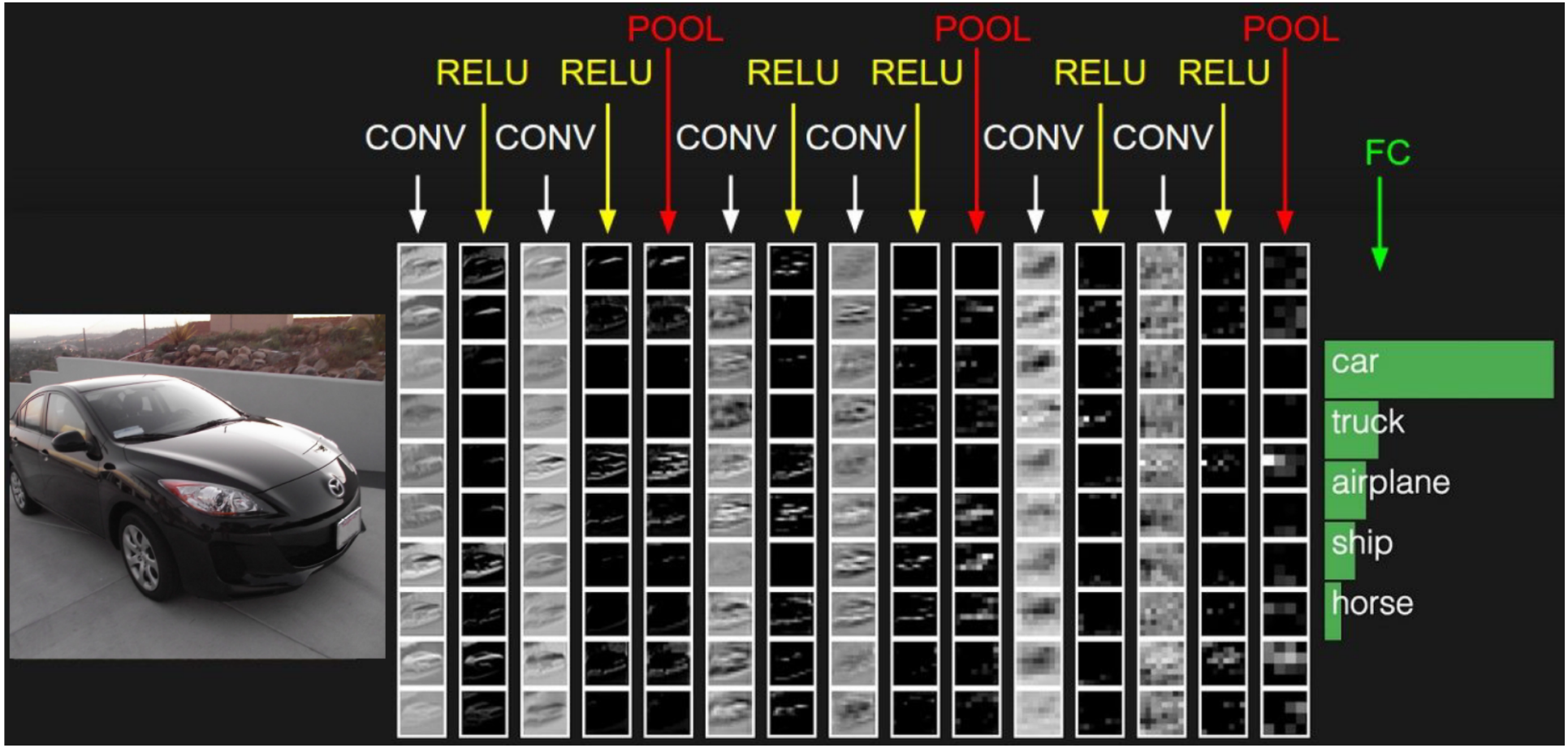
Andrew Ng: Deep Learning, Self-Taught Learning and Unsupervised Feature Learning

Convolutional Neural Network (CNN)



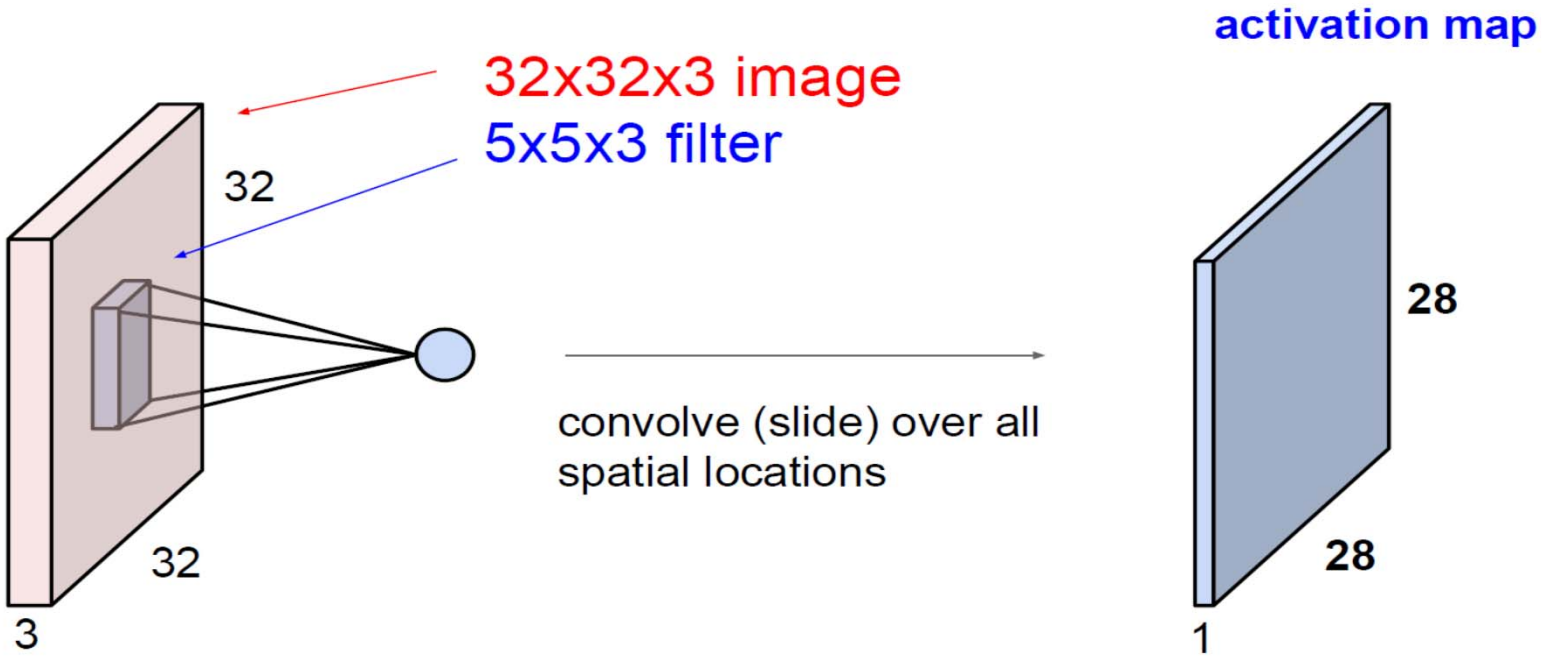
(Image courtesy: http://www.iro.umontreal.ca/~bengioy/talks/DL-Tutorial-NIPS2015.pdf)

Convolutional Neural Network (CNN)



(Courtesy: Stanford course cs231n,http://cs231n.stanford.edu/)

Convolution Layer



An activation map is a 28x28 sheet of neuron outputs:

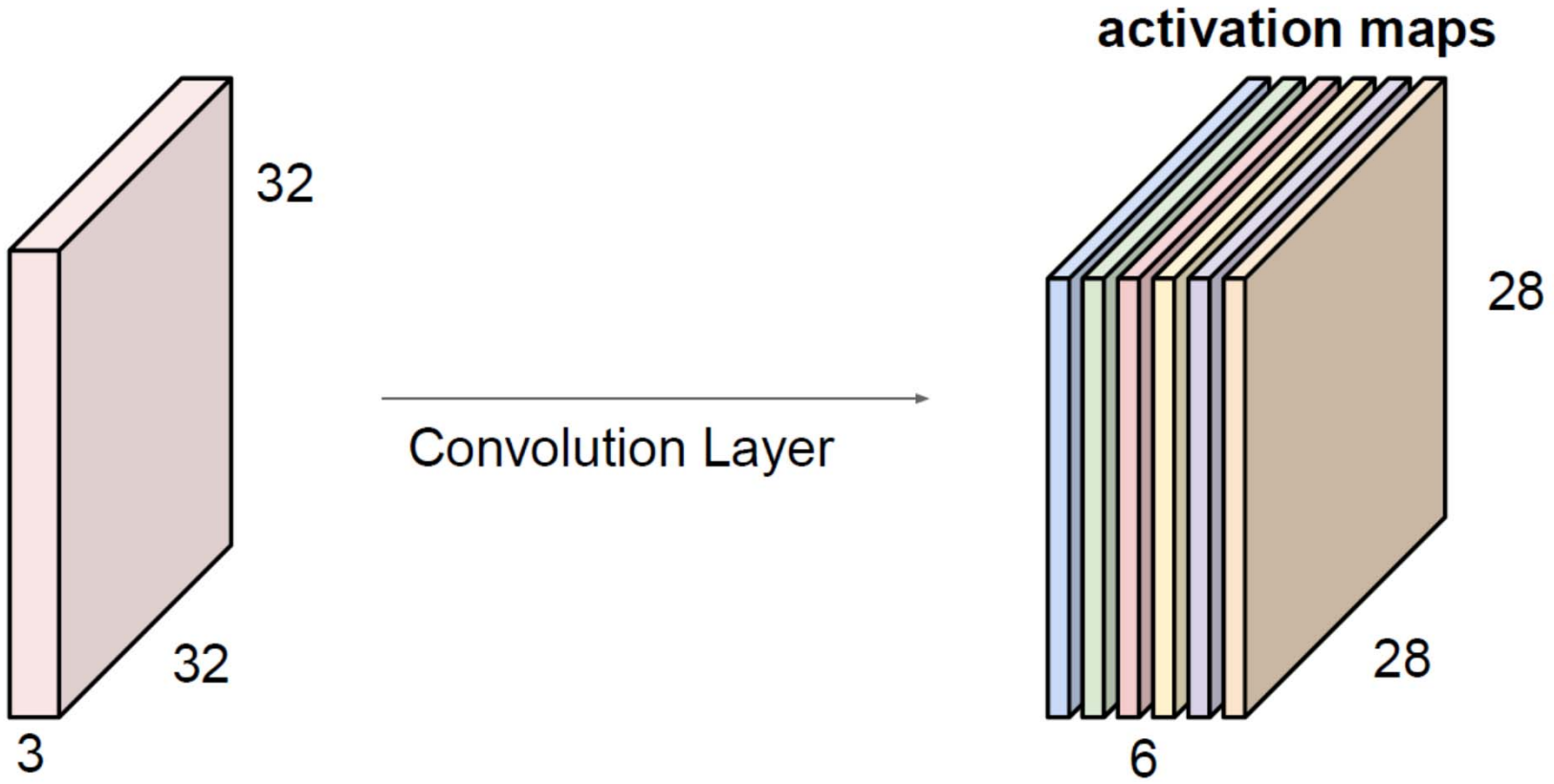
1. Each is connected to a small region in the input
2. All of them share parameters

"5x5 filter" -> "5x5 receptive field for each neuron"

(Courtesy: Stanford course cs231n,http://cs231n.stanford.edu/)

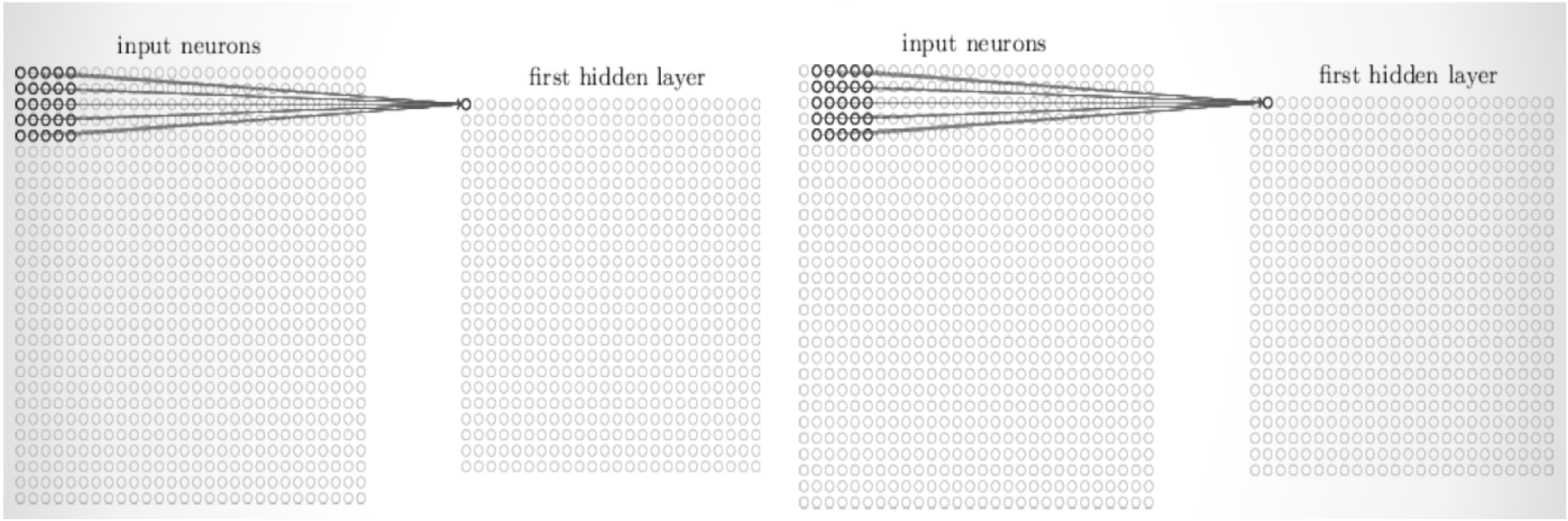
Convolution Layer

If we have 6 5*5 filters, we will get 6 separate activate maps, ie, a new "image" of size 28*28*6



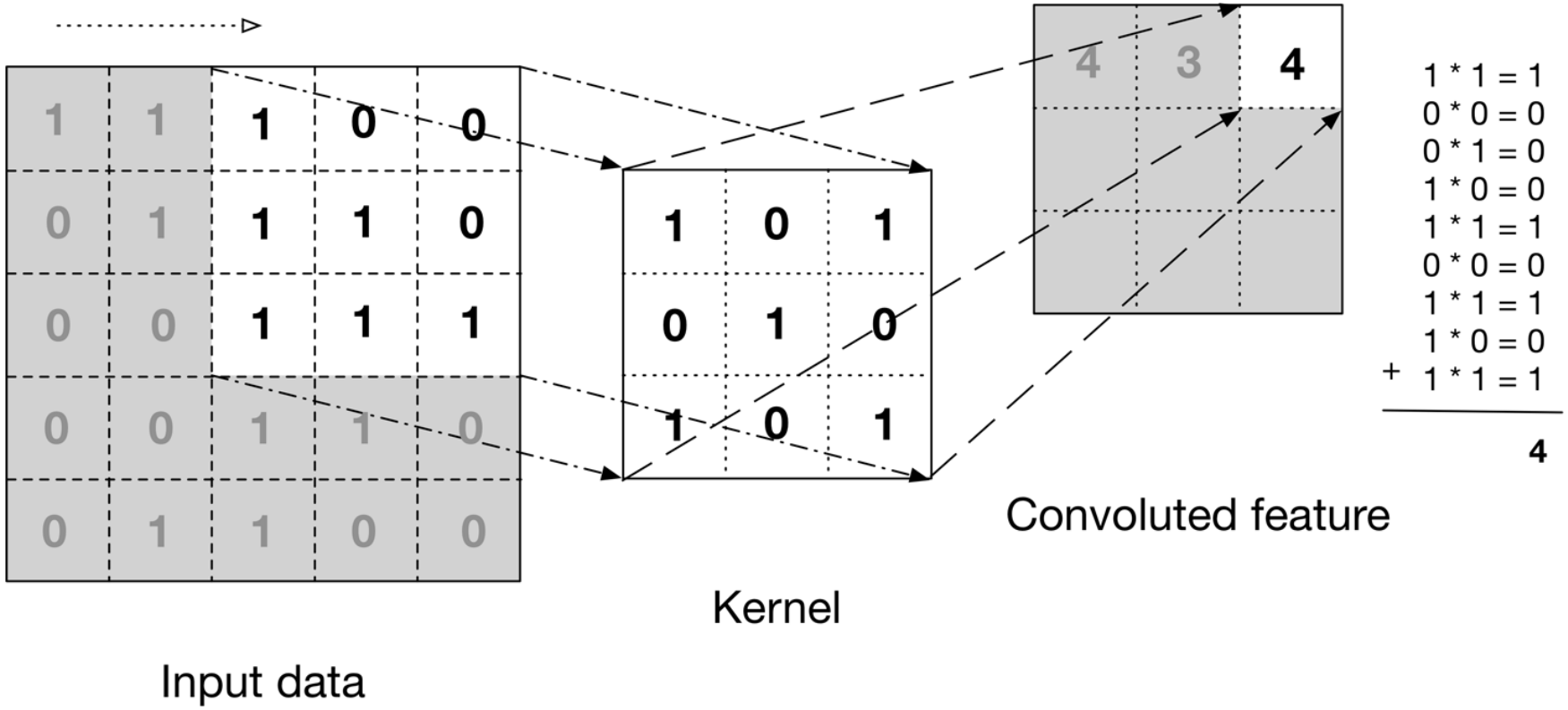
(Courtesy: Stanford course cs231n,http://cs231n.stanford.edu/)

Convolution Layer



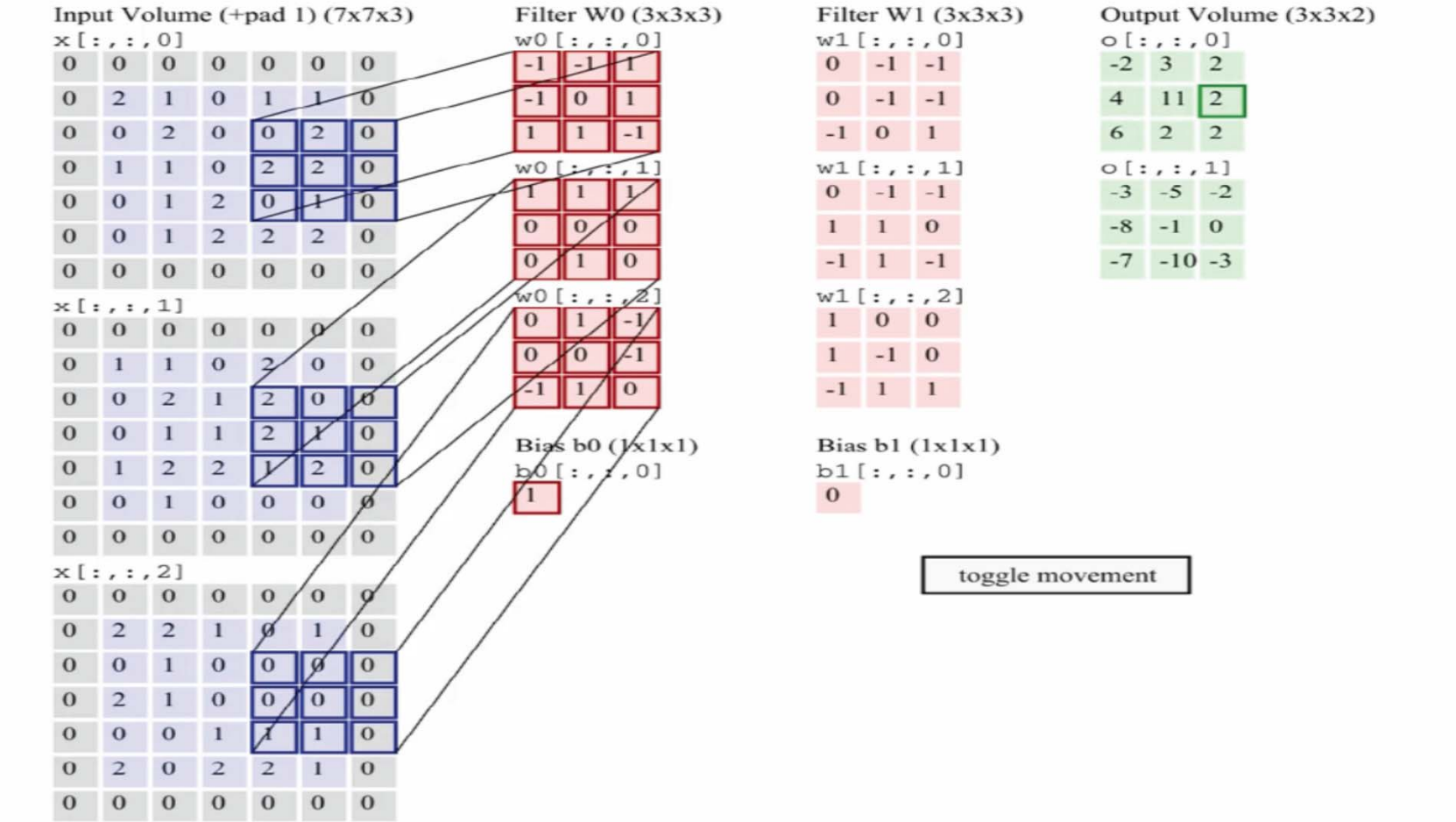
http://neuralnetworksanddeeplearning.com/chap6.html

Convolution Layer



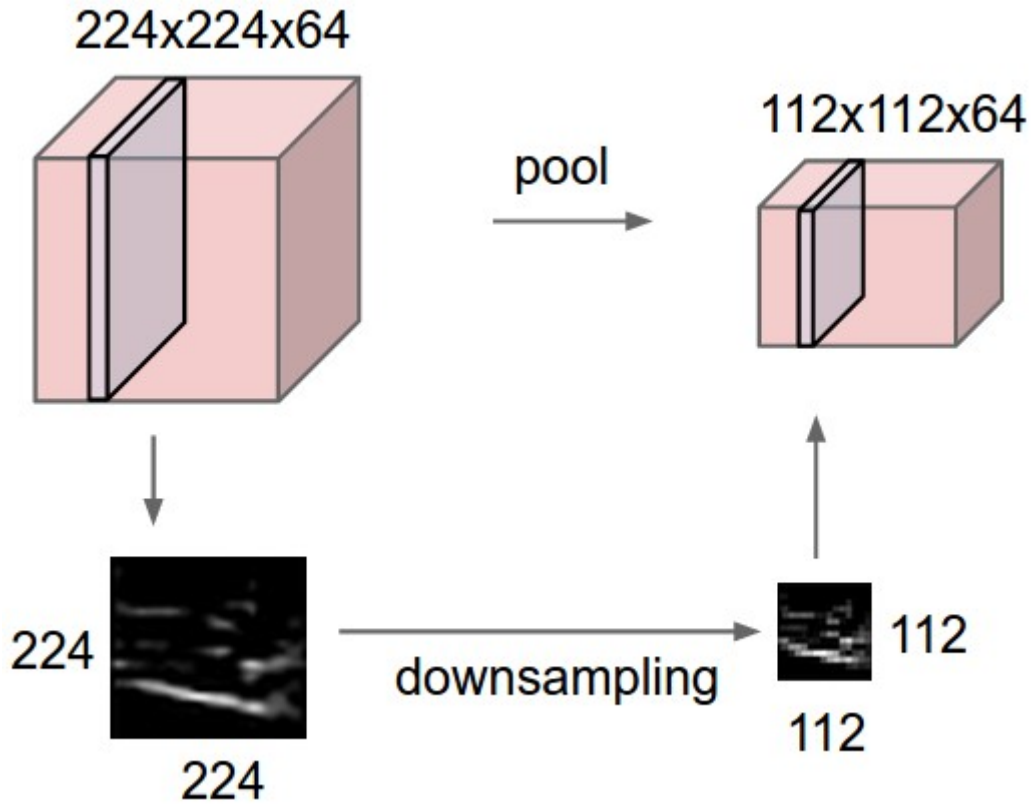
Deep Learning, Adam Gibson, Josh Patterson

Convolution Layer



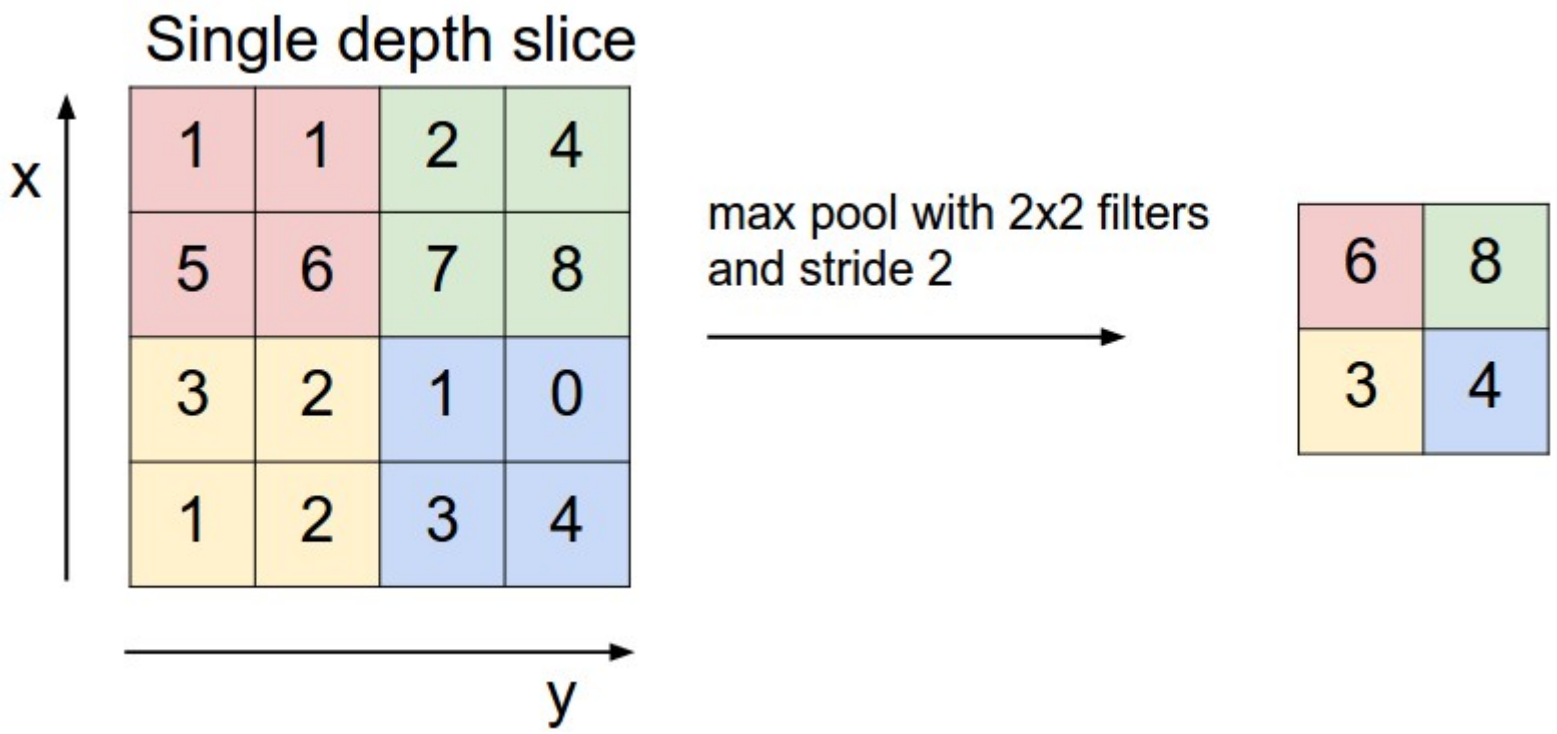
Animation: Andrej Karpathy http://cs231n.github.io/convolutional-networks/

Pooling Layer



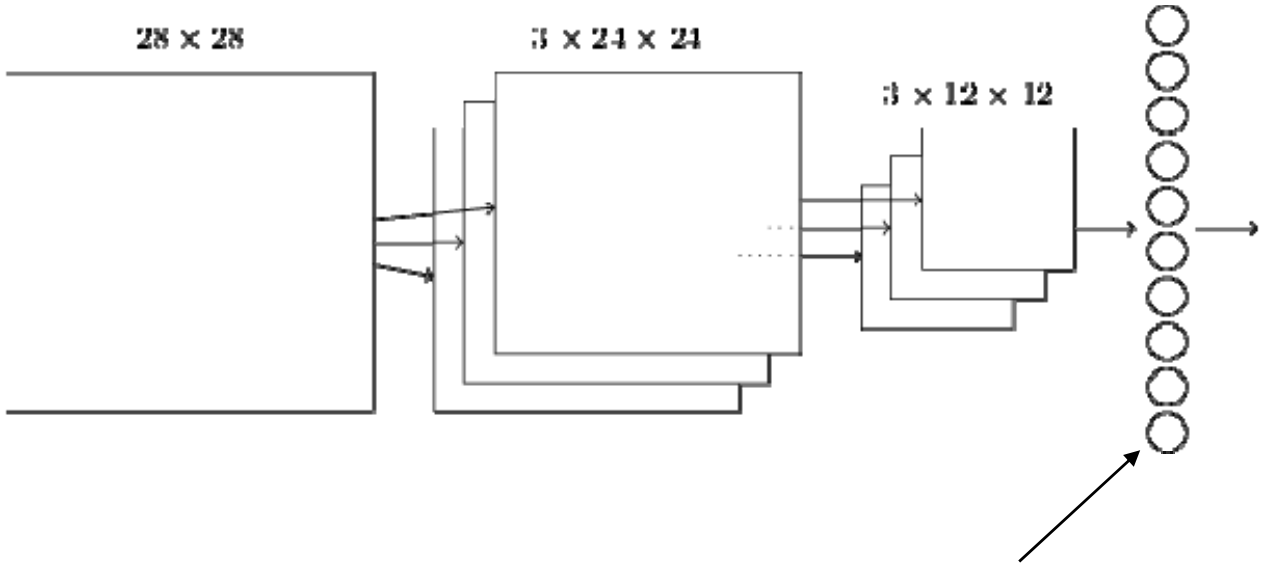
In this example, the input volume of size [224x224x64] is pooled with filter size 2, stride 2 into output volume of size [112x112x64]. Notice that the volume depth is preserved.

Pooling Layer



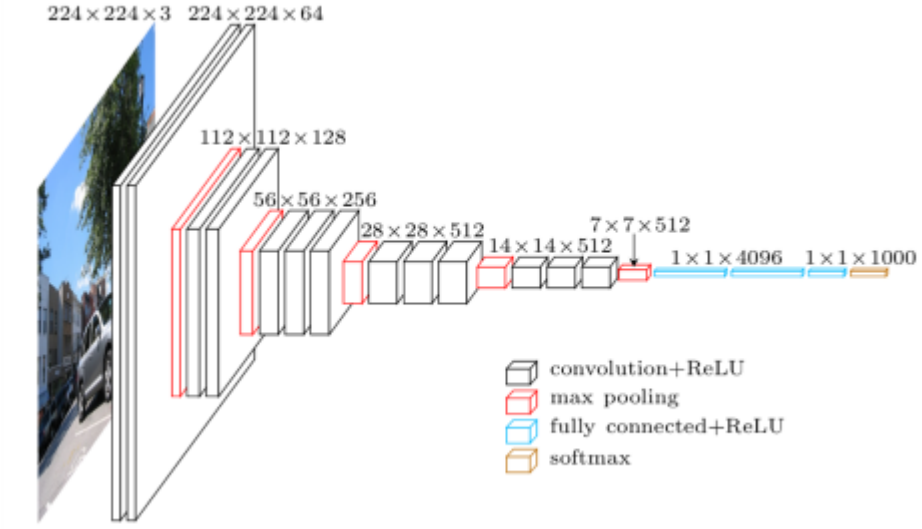
The most common downsampling operation is max, giving rise to max pooling, here shown with a stride of 2. That is, each max is taken over 4 numbers (little 2x2 square).

Fully Connected (FC) Layer



This layer connects *every* neuron from the max-pooled layer to every one of the output neurons.

VGGNet



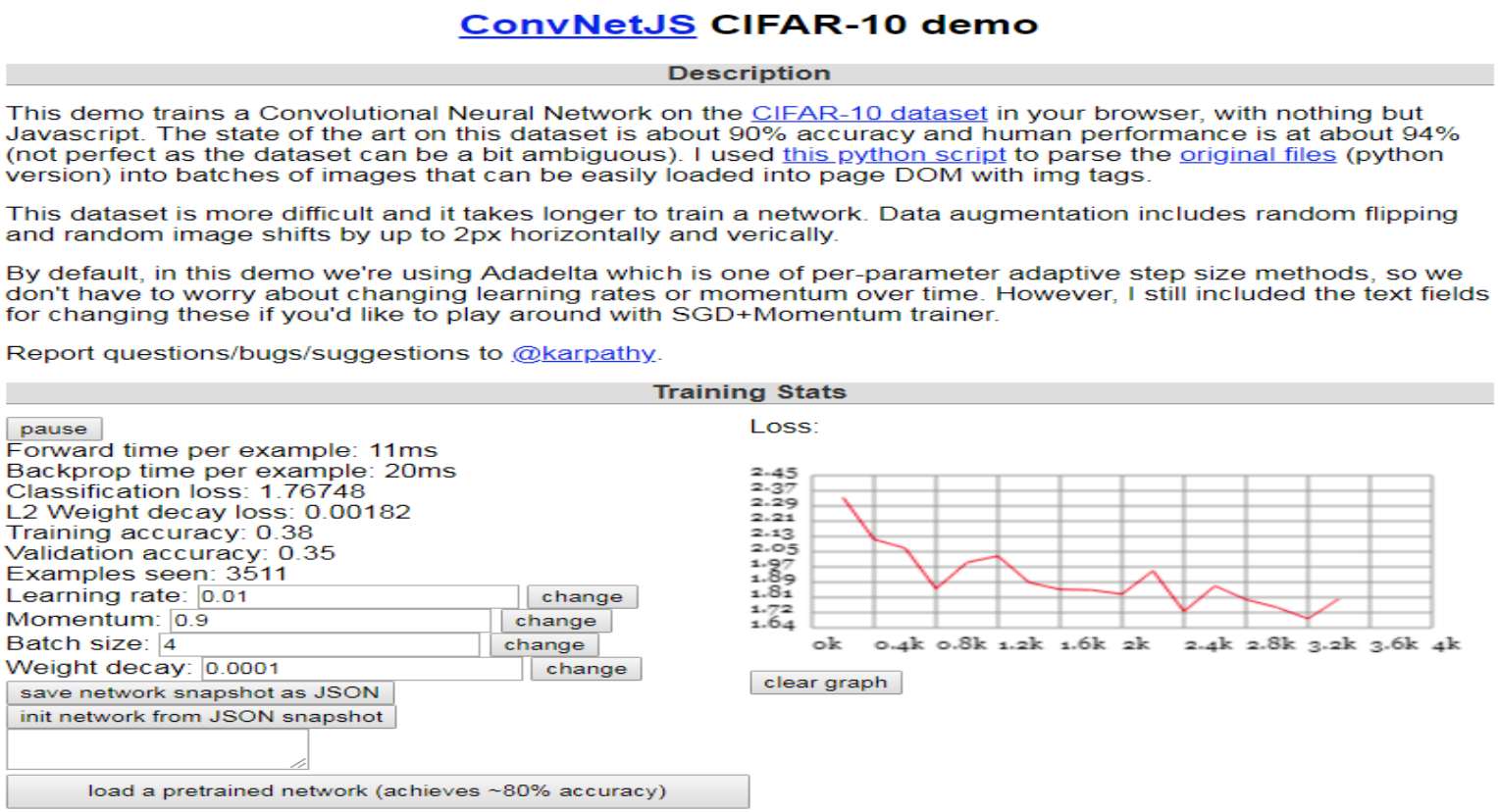
The whole VGGNet is composed of CONV layers that perform 3x3 convolutions with stride 1 and pad 1, and of POOL layers that perform 2x2 max pooling with stride 2 (and no padding).

VGGNet

```
INPUT: [224x224x3]        memory:  224*224*3=150K   weights: 0
CONV3-64: [224x224x64]  memory:  224*224*64=3.2M   weights: (3*3*3)*64 = 1,728
CONV3-64: [224x224x64]  memory:  224*224*64=3.2M   weights: (3*3*64)*64 = 36,864
POOL2: [112x112x64]  memory:  112*112*64=800K   weights: 0
CONV3-128: [112x112x128]  memory:  112*112*128=1.6M   weights: (3*3*64)*128 = 73,728
CONV3-128: [112x112x128]  memory:  112*112*128=1.6M   weights: (3*3*128)*128 = 147,456
POOL2: [56x56x128]  memory:  56*56*128=400K   weights: 0
CONV3-256: [56x56x256]  memory:  56*56*256=800K   weights: (3*3*128)*256 = 294,912
CONV3-256: [56x56x256]  memory:  56*56*256=800K   weights: (3*3*256)*256 = 589,824
CONV3-256: [56x56x256]  memory:  56*56*256=800K   weights: (3*3*256)*256 = 589,824
POOL2: [28x28x256]  memory:  28*28*256=200K   weights: 0
CONV3-512: [28x28x512]  memory:  28*28*512=400K   weights: (3*3*256)*512 = 1,179,648
CONV3-512: [28x28x512]  memory:  28*28*512=400K   weights: (3*3*512)*512 = 2,359,296
CONV3-512: [28x28x512]  memory:  28*28*512=400K   weights: (3*3*512)*512 = 2,359,296
POOL2: [14x14x512]  memory:  14*14*512=100K   weights: 0
CONV3-512: [14x14x512]  memory:  14*14*512=100K   weights: (3*3*512)*512 = 2,359,296
CONV3-512: [14x14x512]  memory:  14*14*512=100K   weights: (3*3*512)*512 = 2,359,296
CONV3-512: [14x14x512]  memory:  14*14*512=100K   weights: (3*3*512)*512 = 2,359,296
POOL2: [7x7x512]  memory:  7*7*512=25K  weights: 0
FC: [1x1x4096]  memory:  4096  weights: 7*7*512*4096 = 102,760,448
FC: [1x1x4096]  memory:  4096  weights: 4096*4096 = 16,777,216
FC: [1x1x1000]  memory:  1000 weights: 4096*1000 = 4,096,000

TOTAL memory: 24M * 4 bytes ~= 93MB / image (only forward! ~*2 for bwd)
TOTAL params: 138M parameters
```

Demo



https://cs.stanford.edu/people/karpathy/convnetjs/demo/cifar10.html

CNN Summary

- **Convolutional Neural Networks (CNN) stack CONV, POOL, FC layers.**
- **Trends toward smaller filter and deeper architectures.**
- **ResNet, GoogleNet… challenge the conventional architectures.**