Convolutional Neural Network: Module 1 - Theory

Artificial Intelligence

Zool Hilmi Ismail
Tokyo City University

Kasparov has Deep Blues after losing

Chess champ: I was rooked

By MICHELE McPHEE, K.C. BAKER and CORKY SIEMASZKO
Daily News Staff Writers

The world's greatest human chess player threw a tantrum and cried foul yesterday after being thrashed by a supercomputer.

It took IBM's Deep Blue just 19 moves to defeat world chess champion Garry Kasparov — a stunning finale to an epic week-long battle of man versus machine.

Not mollified by his $400,000 loser's share, Kasparov stormed off like a sore loser after resigning. He later accused IBM of unfairly programing the high-speed computer to beat him specifically.

He suggested that Deep Blue, which was supposed to play on its own, was coached during the match.

He stopped short of saying the computer team cheated.

"I suspect there were things in the match that were well beyond my understanding," Kasparov said. "And when a big corporation with unlimited resources would like to do so, there are many ways to achieve the result, and the result was achieved."

IBM team leader C.J. Tan denied the computer was coached. "Once the clock started, it relied on Deep Blue's system itself," he said.

Kasparov's pal, Michael Khodarkovski, blamed Kasparov's graceless exit on a lack of practice — he said Kasparov had never lost a match.

Kasparov came close to losing to Anatoly Karpov in a 1984-85 championship match that was suspended without a victory on either side.

Kasparov, 34, considered by some chess experts as the greatest player in the history of the game, last year defeated Deep Blue 4-2.

After losing the opening game of the rematch at the Equitable Center in Manhattan, the computer won the second game and fought Kasparov to draws in the next three.

Then yesterday — with a swiftness that stunned the chess world — Deep Blue took advantage of Kasparov's clumsy opening moves and placed him in a no-win situation after less than an hour of play.

Unable to find a way out, Kasparov — playing the black pieces — tipped his king and resigned. He buried his head in his hands and didn't look at IBM's Tan when they shook hands.

The final score was 3½ points for the computer and 2½ points for Kasparov.

Kasparov said he "cracked under the pressure."

"I am ashamed," said Kasparov, who would have won $700,000 if he had beaten the computer.

Patrick Wolff, author of "The Complete Idiot's Guide to Chess," said the world champ "basically cracked."

Kasparov, playing black, used a standard defense known as the "Caro-Kann," forcing white to sacrifice a piece. But for some reason he botched his seventh move and "he became lost," Wolff said.

"This is not a position he wanted to get into," said Ilya Gurevich, a grand master from Manhattan. "It's a pure calculating position where the computer has a big advantage. The computer's strength is tactics."

The computer Kasparov battled was capable of analyzing 200 million positions per second — twice as many positions per second as the IBM model he defeated in Philadelphia last year.

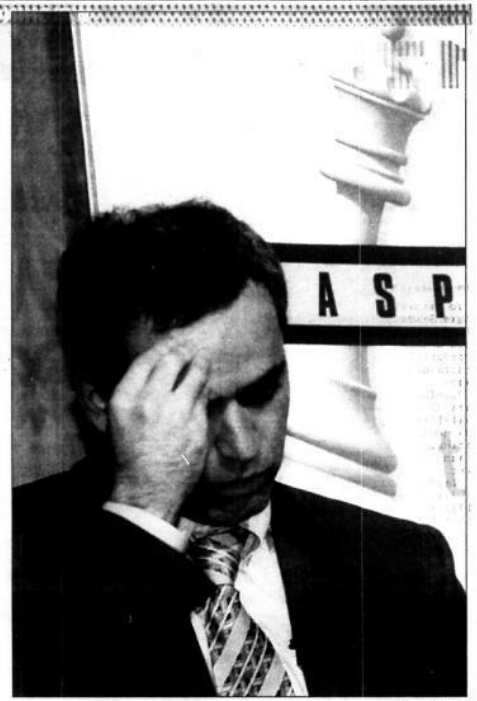
One expert said he was surprised when Kasparov resigned. "It didn't seem lost," said grand master John Fedorowicz of the Bronx, who helped the IBM team prepare its game plan.

At Chess Forum on Thompson St. in Greenwich Village, die-hard chess fans expressed shock at Kasparov's loss.

"This is a historic event," said Mark Wieder, 46, also a computer programer. "The greatest human player of all time lost to a machine."

Chess Forum owner Imad Khachan, 31, said Kasparov was following in the footsteps of other sore losers by suggesting his for didn't play fair.

"This is not uncommon in chess," he said. "When Viktor Korchnoi was playing Karpov in the '70s, Korchnoi made the accusation that the KGB was sending him telepathic messages to destroy his concentration."



LARRY ZUMWALT

RESIGNED: Chess champion Garry Kasparov is disappointed yesterday after losing to IBM's Deep Blue supercomputer. IBM's team leader C.J. Tan (left) denied the computer had been coached, as Kasparov charged after the historic loss in Manhattan. Kasparov will get $400,000 for his efforts.

Artifical intelligence not black and white

Monday, May 12, 1997 • DAILY NEWS

FORGET ABOUT THE Garden or the Meadowlands. The real action was outside the Equitable Center on Seventh Ave., where Garry Kasparov, with a name like a hockey player, did battle with Deep Blue, an IBM supercomputer whose name suggests some starlet who did her best work on 42d St. in those halcyon days before Disney.

The scalpers were asking as much as $500 for a $25 seat. "Actually, I'd settle for a couple of hundred," said Ze Ayala. "I have extras."

In the history of New York, there's never been a scalper so hopelessly well mannered as Ze Ayala, Ph.D. Instead of the usual hawker's cry — "Who got tickets?" — Ayala was content to let the business come to him as he burnished his new henna tattoo with a cotton ball doused in lemon juice. "The lemon juice helps the absorption of the dye," he said. The tattoo bore the name of his band, "Flashpot," for whom the tall, long-haired Ayala plays guitar. It should be noted that in lieu of a day job, he works at the Institute of Molecular Evolutionary Genetics at Penn State. His mission wasn't even mercenary; it was professional. He said he wouldn't have sold his own seat for a million bucks.

"My field is artificial intelligence," he said. "And this is the coolest thing to happen in my lifetime."

In such a spirit, the 28-year-old scientist had come to witness the inevitable coolness: Man mangled by Machine.

He couldn't help but root for Kasparov. The sentimental part of him was taken with the charms of obsolescence. But he knew better than to bet against technology.

Kasparov is the best chess player in the world; but unlike Deep Blue, he can also be vain, angry, neurotic, panicked, fearful, in all, human. "Chess is fundamentally psychological," Ayala said. "And that's precisely what Kasparov has working against him."

I asked him how long before Deep Blue is playing lead guitar in his band.

That day will come, he said, and it won't be too long. "You know the band Nine Inch Nails?" he said. "That's all computers. But what you're really asking is: Will a computer be able to write

SEE KRIEGEL PAGE 28

Computer Chess

- 2/96: Kasparov vs Deep Blue
  - Kasparov victorious: 3 wins, 2 draws, 1 loss
- 3/97: Kasparov vs Deeper Blue
  - First match won against world champion
  - 512 processors: 200 million chess positions per second

How do you think it works???

FACTS ON DEEP BLUE

- ✓ **brute force** computing power
- ✓ **massively parallel**, RS/6000 SP Thin P2SC-based system with 30 nodes, with each node containing a 120 MHz P2SC microprocessor,
- ✓ enhanced with **480 special purpose VLSI** chess chips.
- ✓ Its chess playing program was written in C and ran under the **AIX operating system**.
- ✓ capable of evaluating **200 million positions per second**, twice as fast as the 1996 version.
- ✓ In June 1997, Deep Blue was the 259th most powerful supercomputer according to the TOP500 list, achieving **11.38 GFLOPS** on the High-Performance LINPACK benchmark.

AIX is an open **operating system** from IBM that is based on a version of UNIX

CHESS PLAYER DEFEATED???

NEXT GAMEBOARD???

AlphaGo

- ✓ developed by **Google DeepMind in London** to play the board game Go.
- ✓ In **October 2015,** it became the first Computer Go program to beat a professional human Go player
- ✓ In **March 2016,** it beat Lee Sedol in a five-game match, the first time a computer Go program has beaten a 9-dan professional without handicaps.

HOW IT WORKS???

FACTS OF ALPHAGO

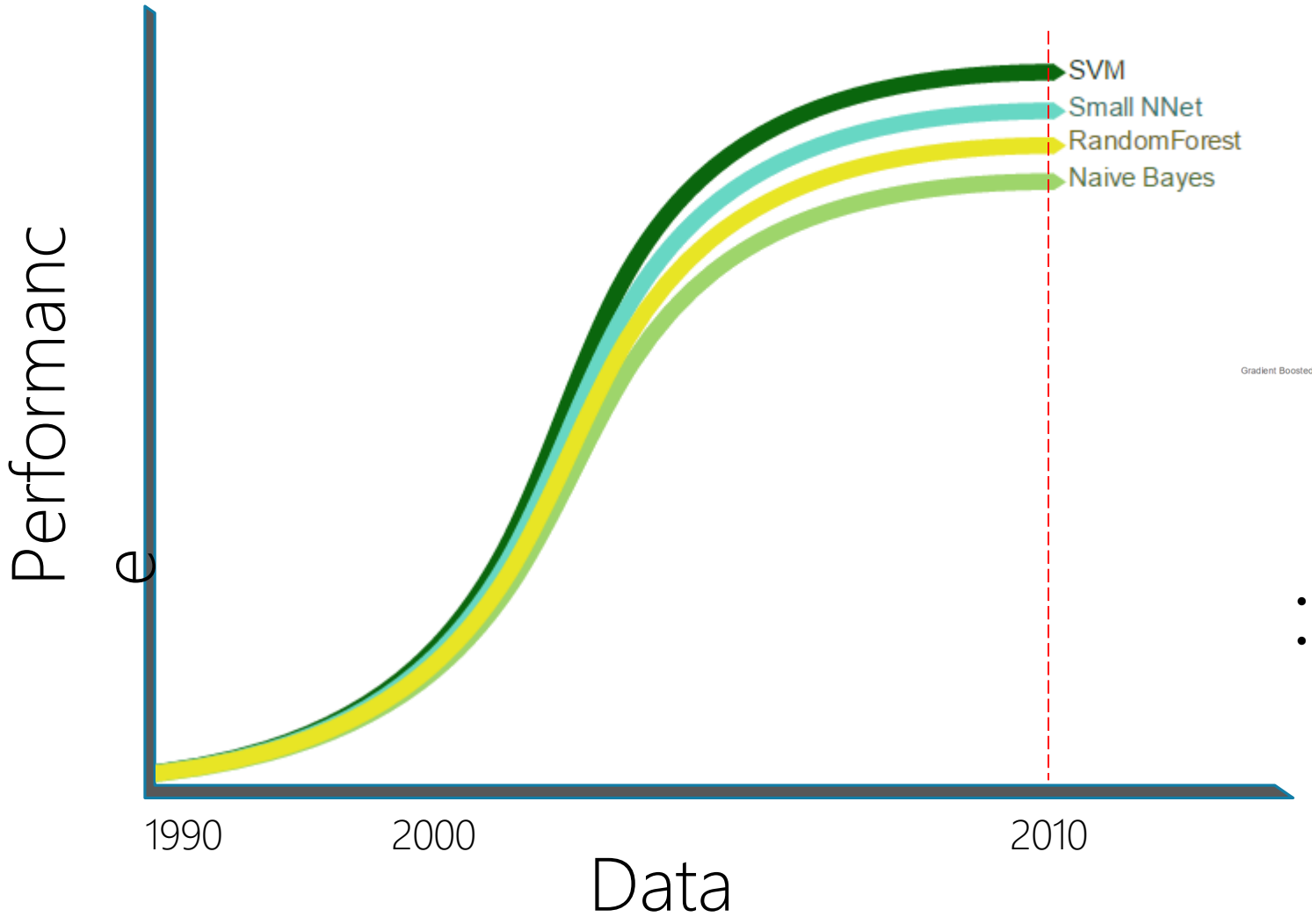
The Elo rating system is a method for calculating the relative skill levels of players in competitor-versus-competitor games such as chess.

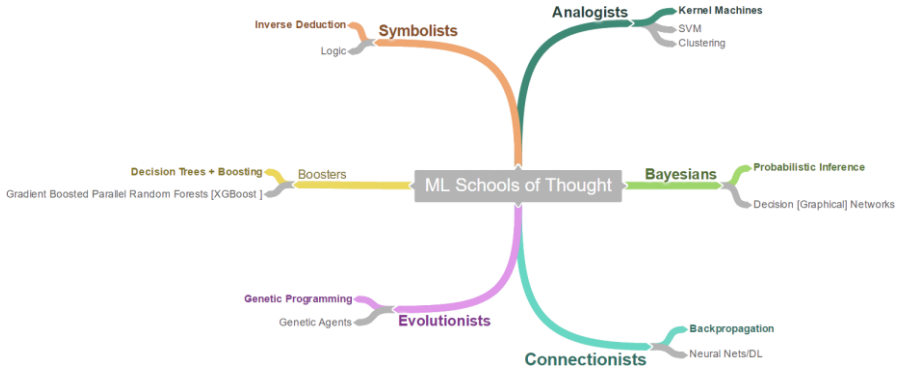
In the paper, written in 2015, the strength of various AIs was estimated as follows:

AI name	Elo rating
Distributed AlphaGo (2015)	3140
AlphaGo (2015)	2890
CrazyStone	1929
Zen	1888
Pachi	1298
Fuego	1148
GnuGo	431

AlphaGo ran on 48 CPUs and 8 GPUs and the distributed version of AlphaGo ran on 1202 CPUs and 176 GPUs.

TREND #1 [SCALE]

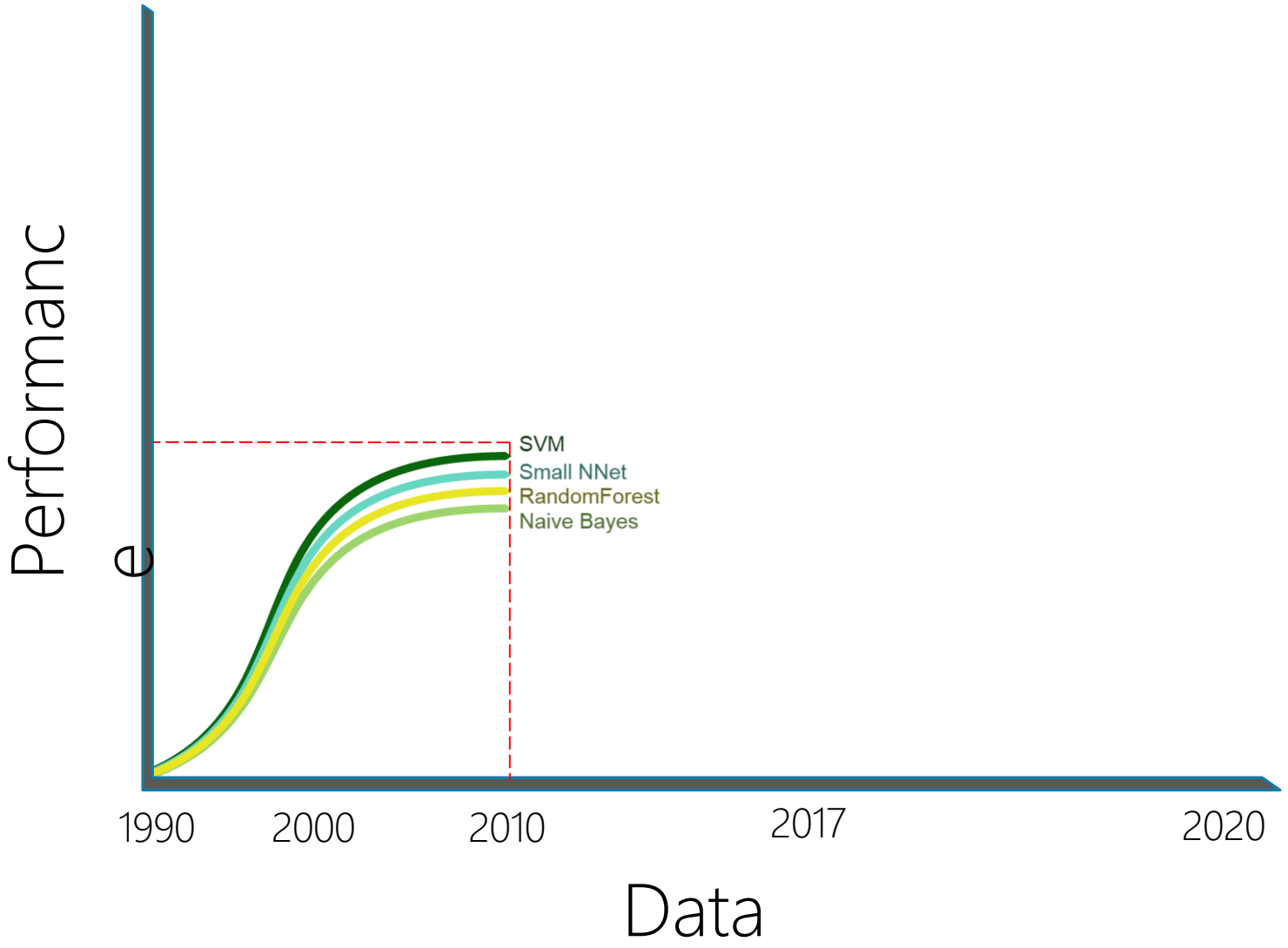




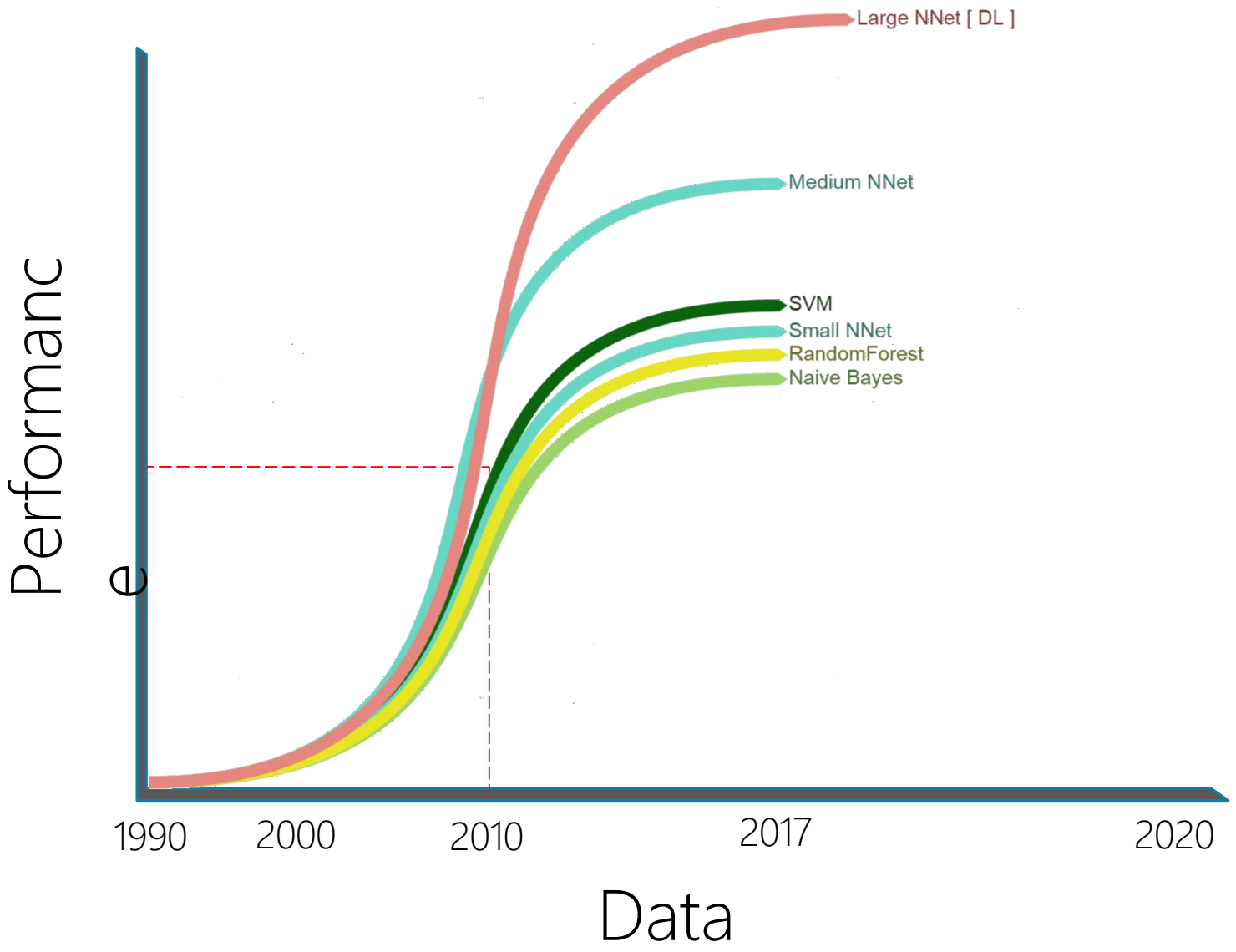
- Exponential Growth of Data
- Somewhat Arbitrary Ordering of Model Performance

  [based on hand engineering effort]

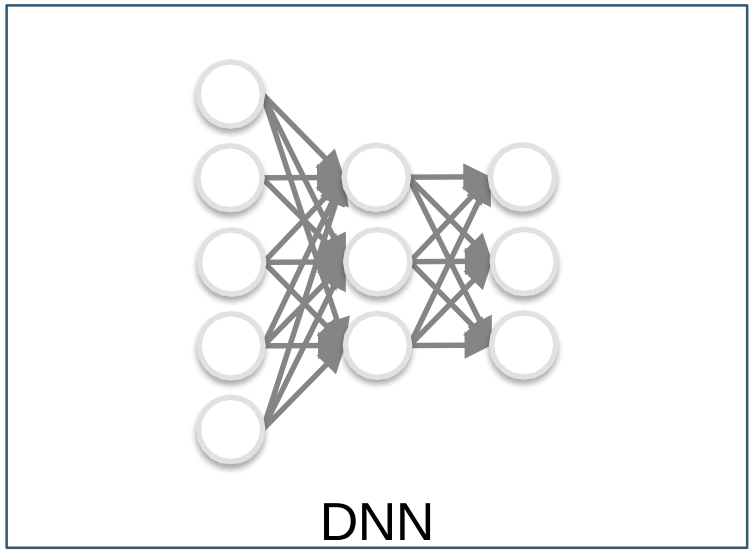
TREND #1 [SCALE]



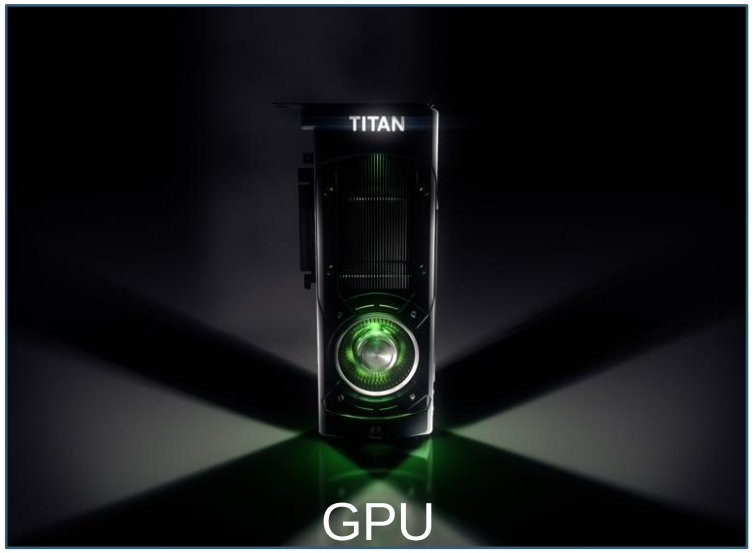
TREND #1 [SCALE]



THE BIG BANG IN MACHINE LEARNING



DNN



GPU

BIG DATA

WIRED

HISTORY

Yann LeCun, Professor of Computer Science
The Courant Institute of Mathematical Sciences
New York University
Room 1220, 715 Broadway, New York, NY 10003, USA.
(212)998-3283 yann@cs.nyu.edu

- In 1995, Yann LeCun and Yoshua Bengio introduced the concept of convolutional neural networks.

ABOUT CNN'S

- **Convolutional neural network** (**CNN,** or **ConvNet**) is a class of deep, feed-forward artificial neural network that have successfully been applied to analyze visual imagery.

- CNN's Were neurobiologically motivated by the findings of locally sensitive and orientation-selective nerve cells in the visual cortex.

- They designed a network structure that implicitly extracts relevant features.

- Convolutional Neural Networks are a special kind of multi-layer neural networks

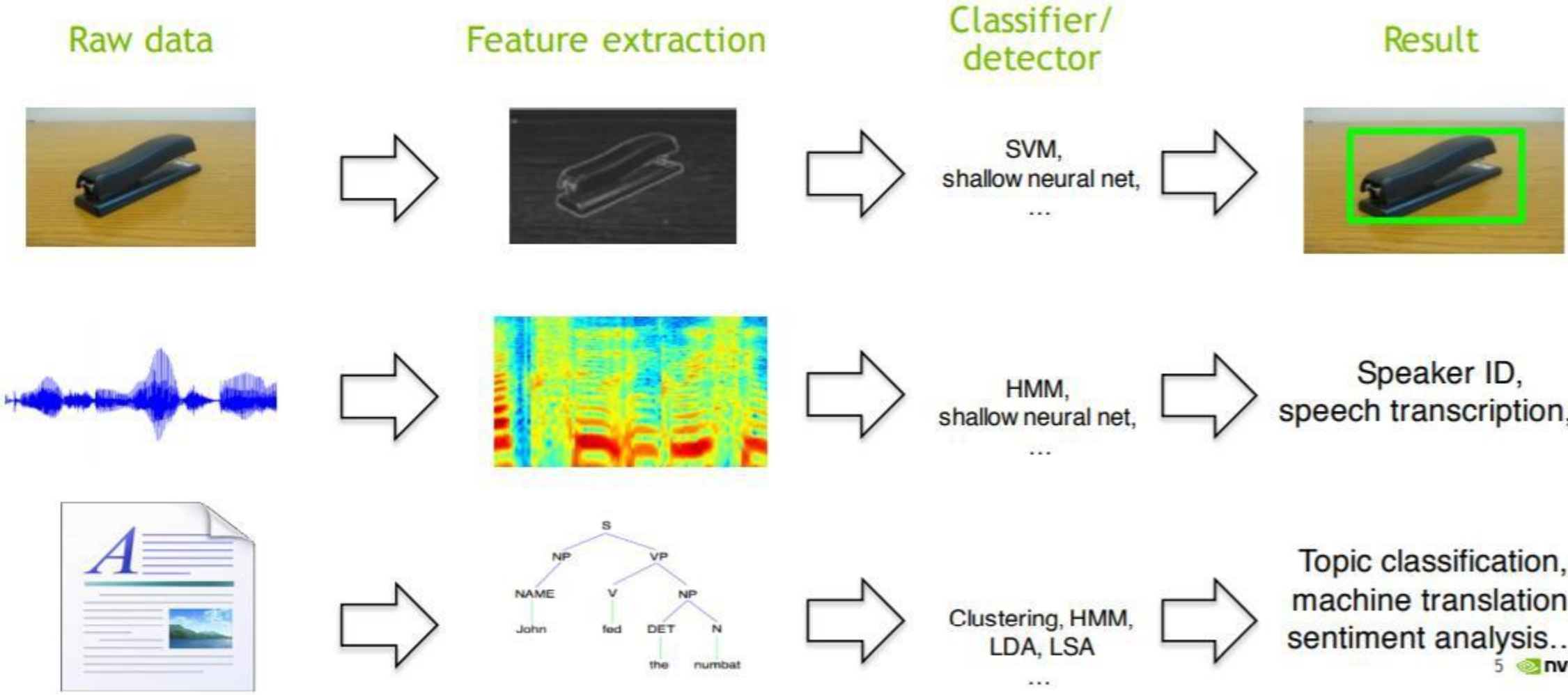
CONVOLUTIONAL NEURAL NETWORKS

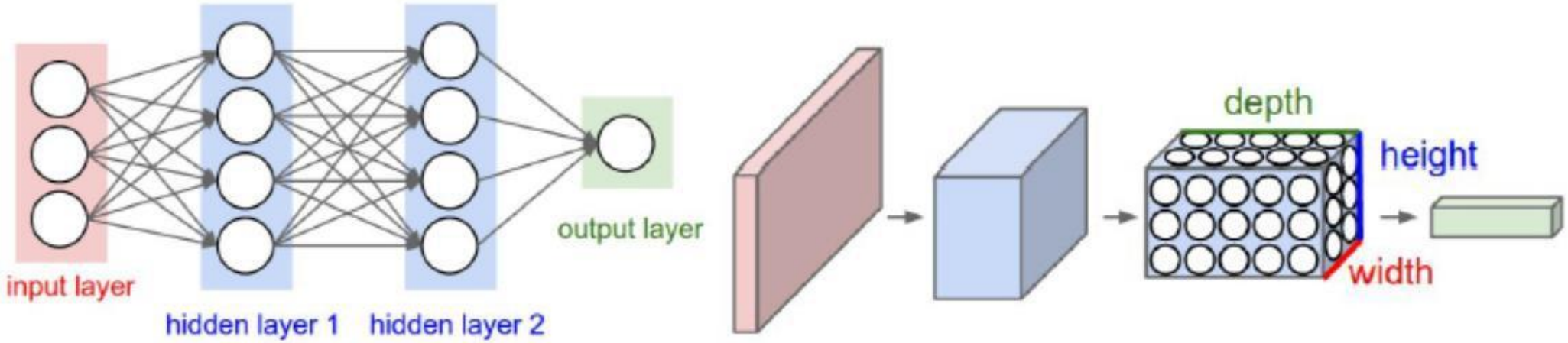
- Biologically inspired.
- Neuron only connected to a small region of neurons in layer below it
  called the *receptive field*.
- A given layer can have many convolutional filters/kernels.
  Each filter has the same weights across the whole layer.
- Bottom layers are convolutional, top layers are fully connected.
- Generally trained via supervised learning.

  Supervised
  Unsupervised
  Reinforcement ...ideal system automatically switches modes...

Traditional machine perception

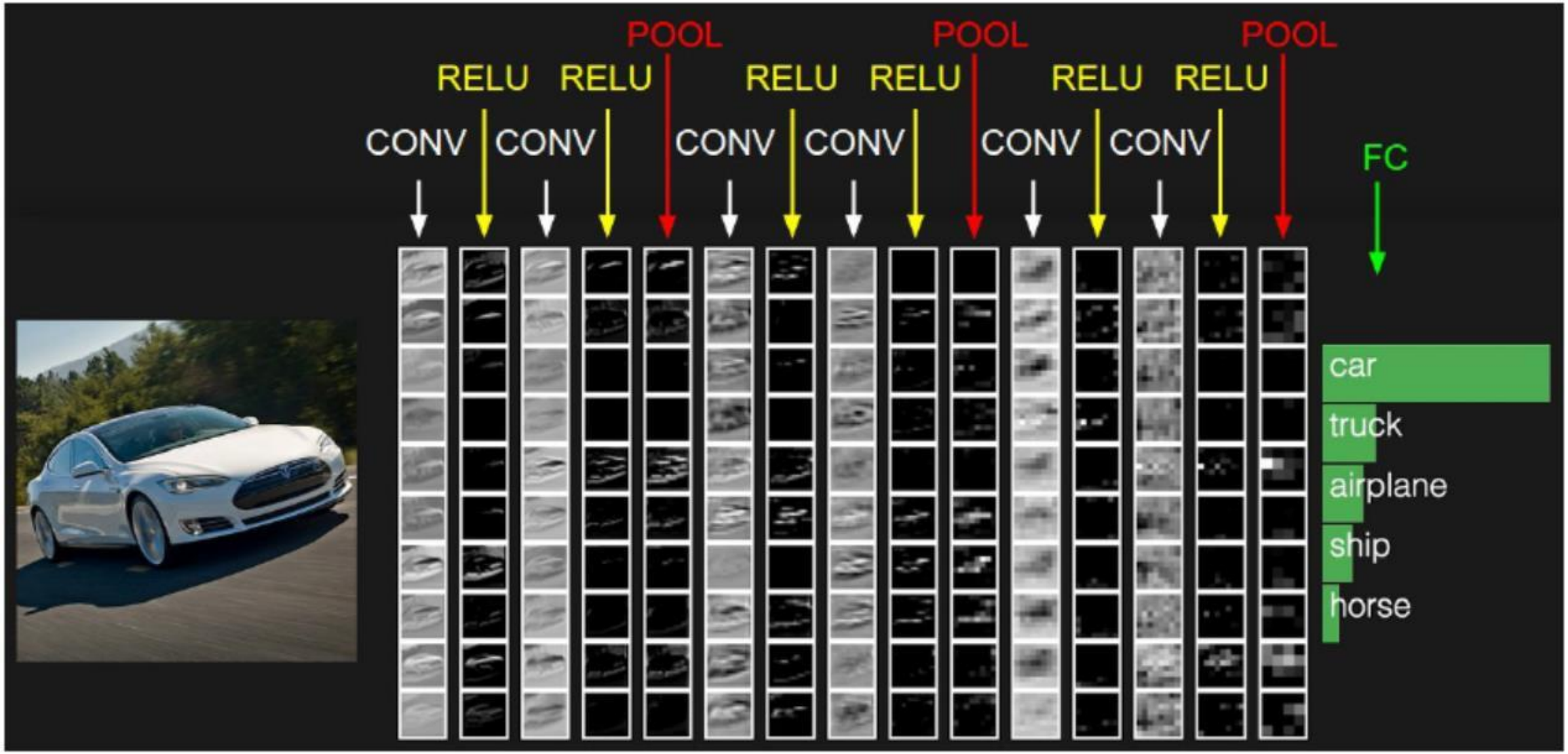
Hand crafted feature extractors





Left: A regular 3-layer Neural Network. Right: A ConvNet arranges its neurons in three dimensions (width, height, depth), as visualized in one of the layers. Every layer of a ConvNet transforms the 3D input volume to a 3D output volume of neuron activations. In this example, the red input layer holds the image, so its width and height would be the dimensions of the image, and the depth would be 3 (Red, Green, Blue channels).

CNN TOPOLOGY

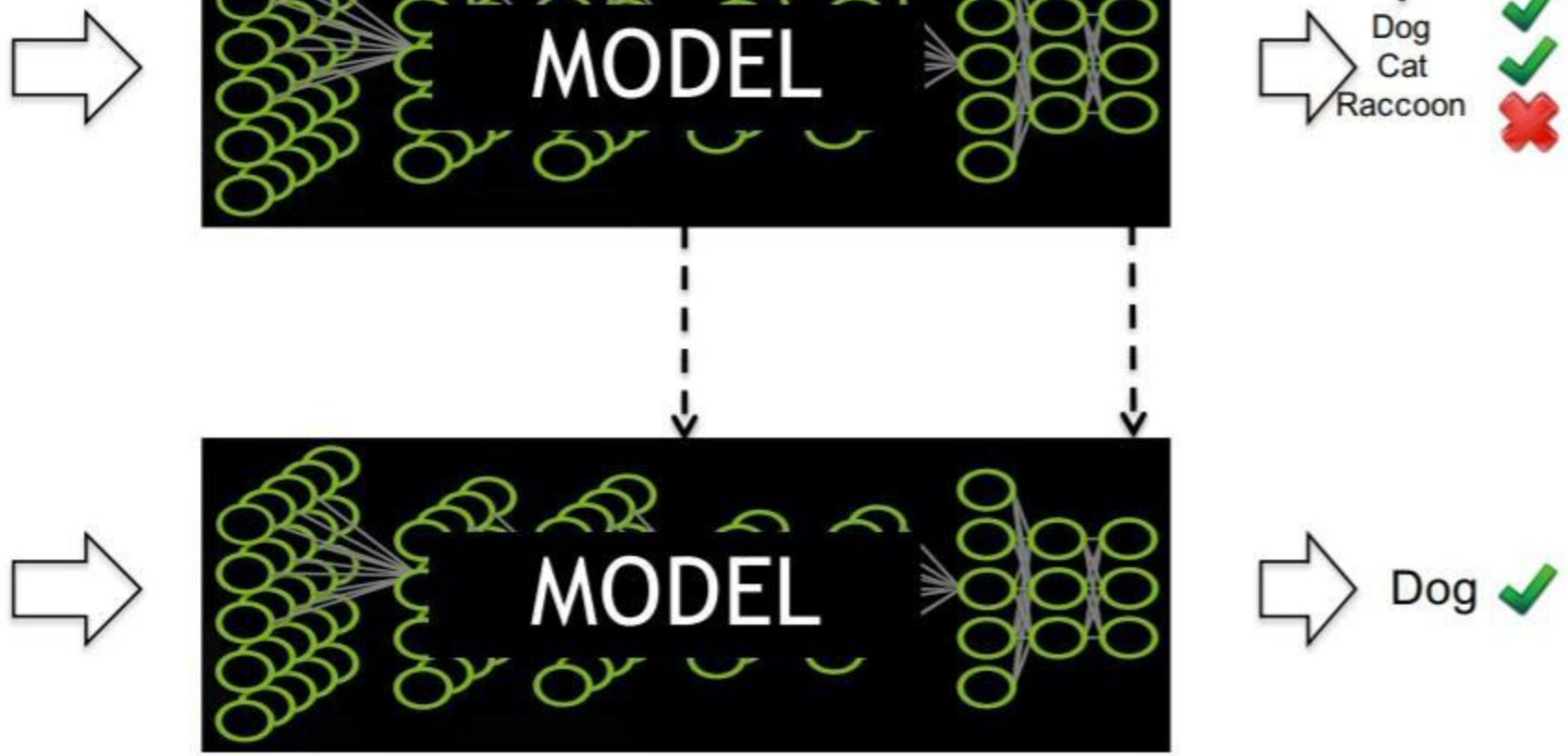


Certainly, coming up with features is difficult, time-consuming and requires expert knowledge.

A lot of time is spend tuning the features which are often hand-crafted!

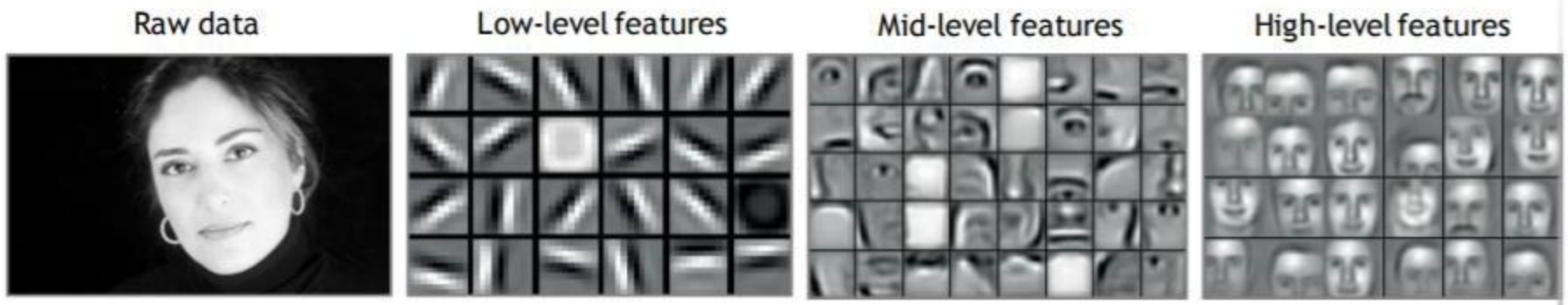
Deep learning approach

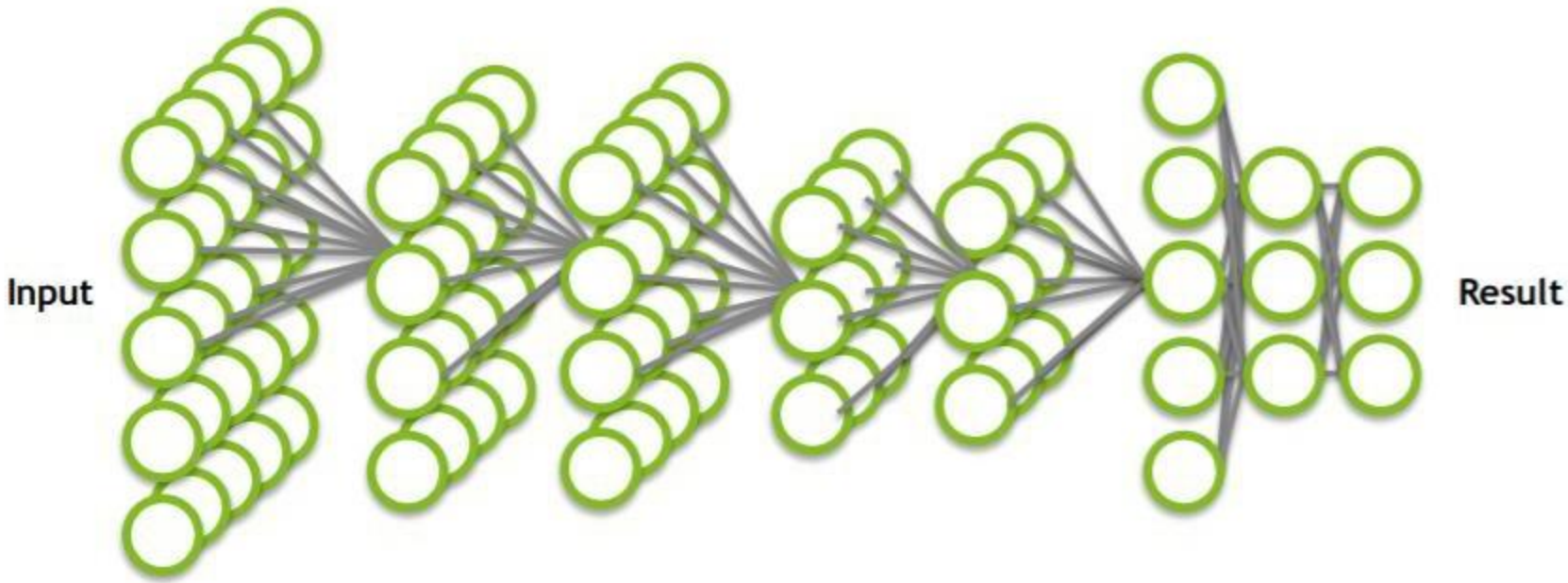
Train:



Deploy:

Deep neural network (dnn)





Application components:

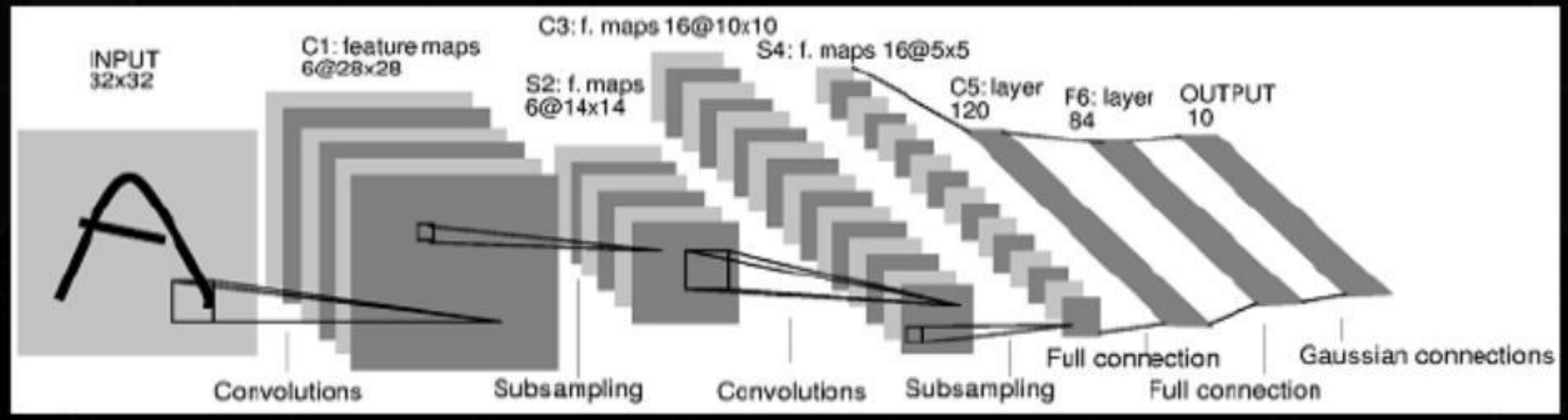
Task objective
- e.g. Identify face

Training data
- 10-100M images

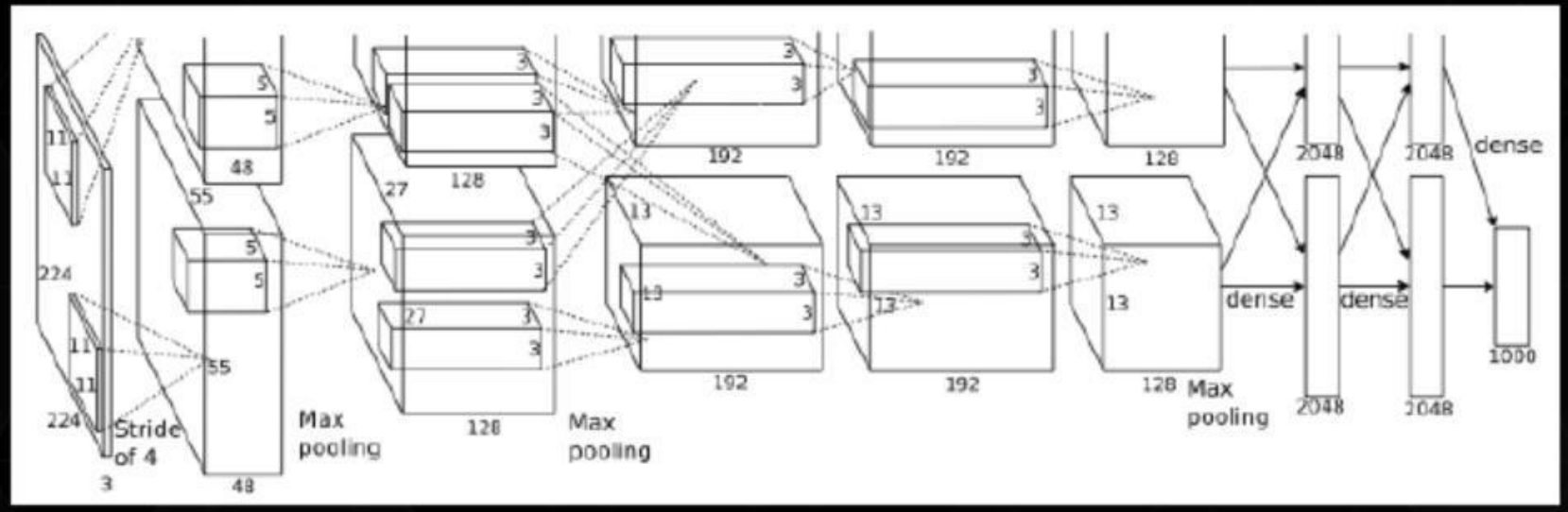
Network architecture
- ~10 layers
- 1B parameters

Learning algorithm
- ~30 Exaflops
- ~30 GPU days

CONVOLUTIONAL NETWORKS BREAKTHROUGH



Y. LeCun et al. 1989-1998 : Handwritten digit reading



A. Krizhevsky, G. Hinton et al. 2012 : Imagenet classification winner

CONVOLUTIONS – THE MAIN WORKLOAD

- Very compute intensive, but with a large parameter space

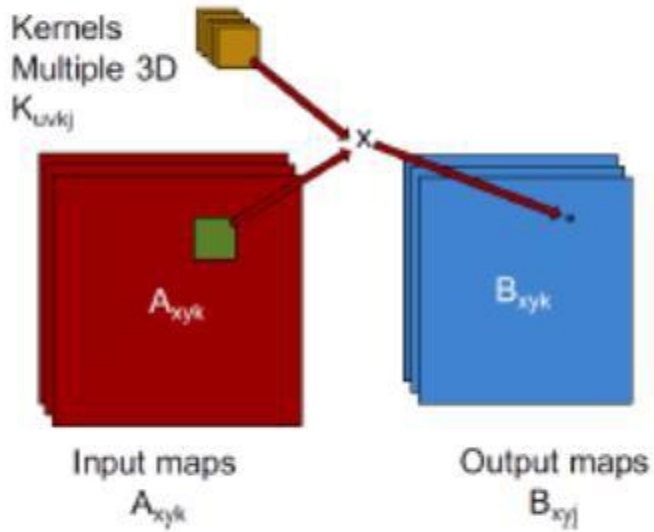
1. Minibatch Size
2. Input feature maps
3. Image Height
4. Image Width
5. Output feature maps
6. Kernel Height
7. Kernel Width
8. Top zero padding
9. Side zero padding
10. Vertical stride
11. Horizontal stride

- Layout and configuration variations

- Other cuDNN routines have straightforward implementations

CNN

Requires convolution and M x V



Filters conserved through plane

Multiply limited - even without batching.

6D Loop
For each output map j
  For each input map k
    For each pixel x,y
      For each kernel element u,v
        B_{xyj} += $A_{(x-u)(y-v)k}$ x K_{uvkj}

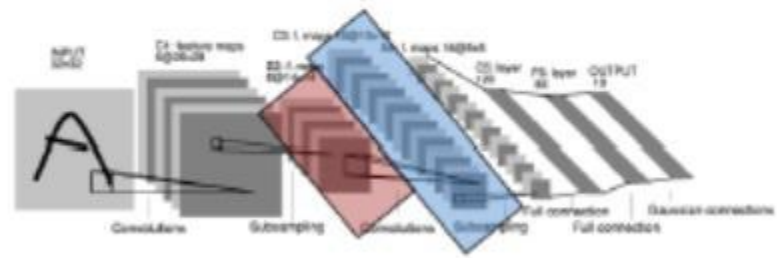


Image Size: [*W* = 5]x[*H* = 5]x [*D* = 3]

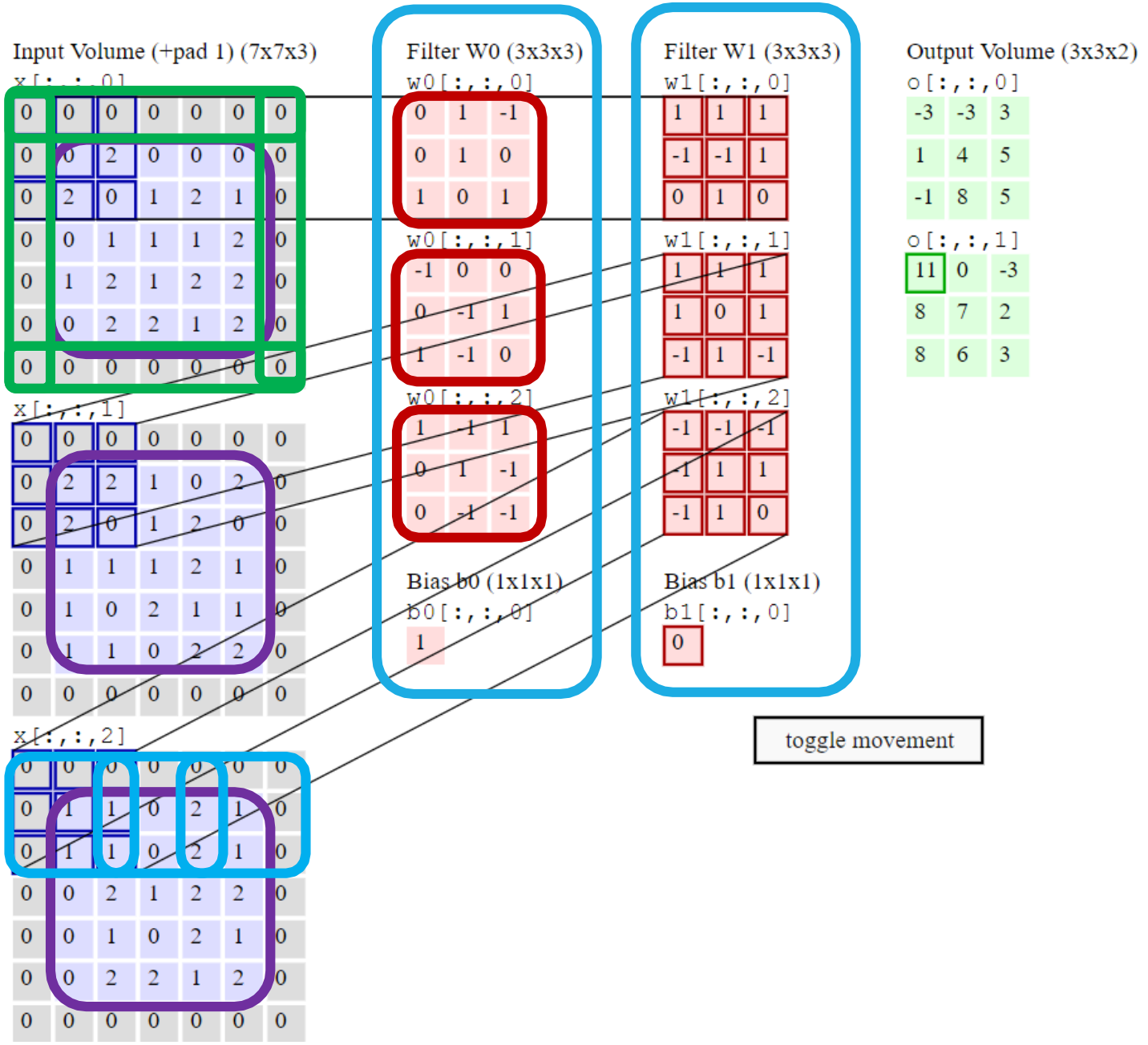
= Number of Filter: *K* = 2

Receptor Field Size:

***F* =3 [[*W* = 3]x[*H* = 3]x [*D* = 3]]**

Padding: *P* = 1

Stride: *S* = 2



Summary. To summarize, the Conv Layer:

- Accepts a volume of size $W_1 \times H_1 \times D_1$
- Requires four hyperparameters:
  - Number of filters K,
  - their spatial extent F,
  - the stride S,
  - the amount of zero padding P.
- Produces a volume of size $W_2 \times H_2 \times D_2$ where:
  - $W_2 = (W_1 - F + 2P)/S + 1$
  - $H_2 = (H_1 - F + 2P)/S + 1$ (i.e. width and height are computed equally by symmetry)
  - $D_2 = K$
- With parameter sharing, it introduces $F \cdot F \cdot D_1$ weights per filter, for a total of $(F \cdot F \cdot D_1) \cdot K$ weights and K biases.
- In the output volume, the d-th depth slice (of size $W_2 \times H_2$) is the result of performing a valid convolution of the d-th filter over the input volume with a stride of S, and then offset by d-th bias.

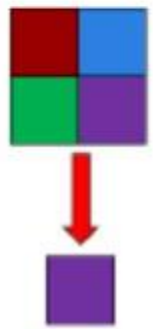
Real-world example. The Krizhevsky et al. architecture that won the ImageNet challenge in 2012 accepted images of size [227x227x3]. On the first Convolutional Layer, it used neurons with receptive field size $F = 11$ stride $S = 4$ and no zero padding $P = 0$. Since (227 - 11)/4 + 1 = 55 and since the Conv layer had a depth of $K = 96$ the Conv layer output volume had size [55x55x96]. Each of the 55*55*96 neurons in this volume was connected to a region of size [11x11x3] in the input volume. Moreover, all 96 neurons in each depth column are connected to the same [11x11x3] region of the input, but of course with different weights. As a fun aside, if you read the actual paper it claims that the input images were 224x224, which is surely incorrect because (224 - 11)/4 + 1 is quite clearly not an integer. This has confused many people in the history of ConvNets and little is known about what happened. My own best guess is that Alex used zero-padding of 3 extra pixels that he does not mention in the paper.

Number of neurons = 55x55x96

Each neuron weight = 11x11x3 + 1 (bias)

Other Operations

To finish building a DNN



Pooling

ReLU
(or other non-linear function)

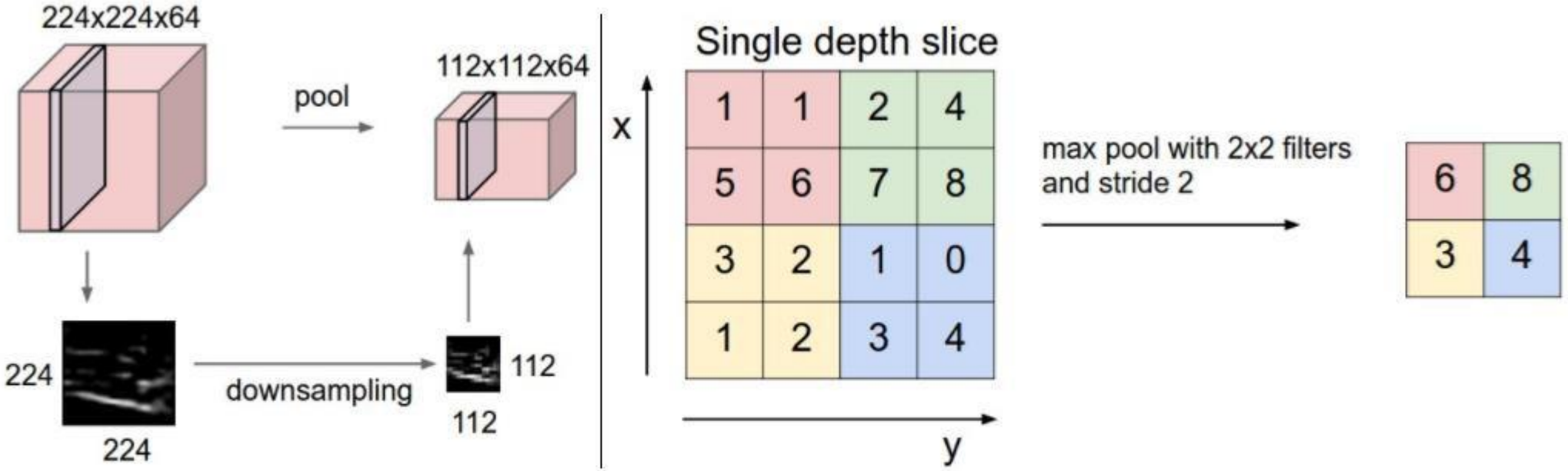
$$w_{ij} \mathrel{+}= \alpha a_j g_i$$

Weight Update

These are not limiting factors with appropriate GPU use

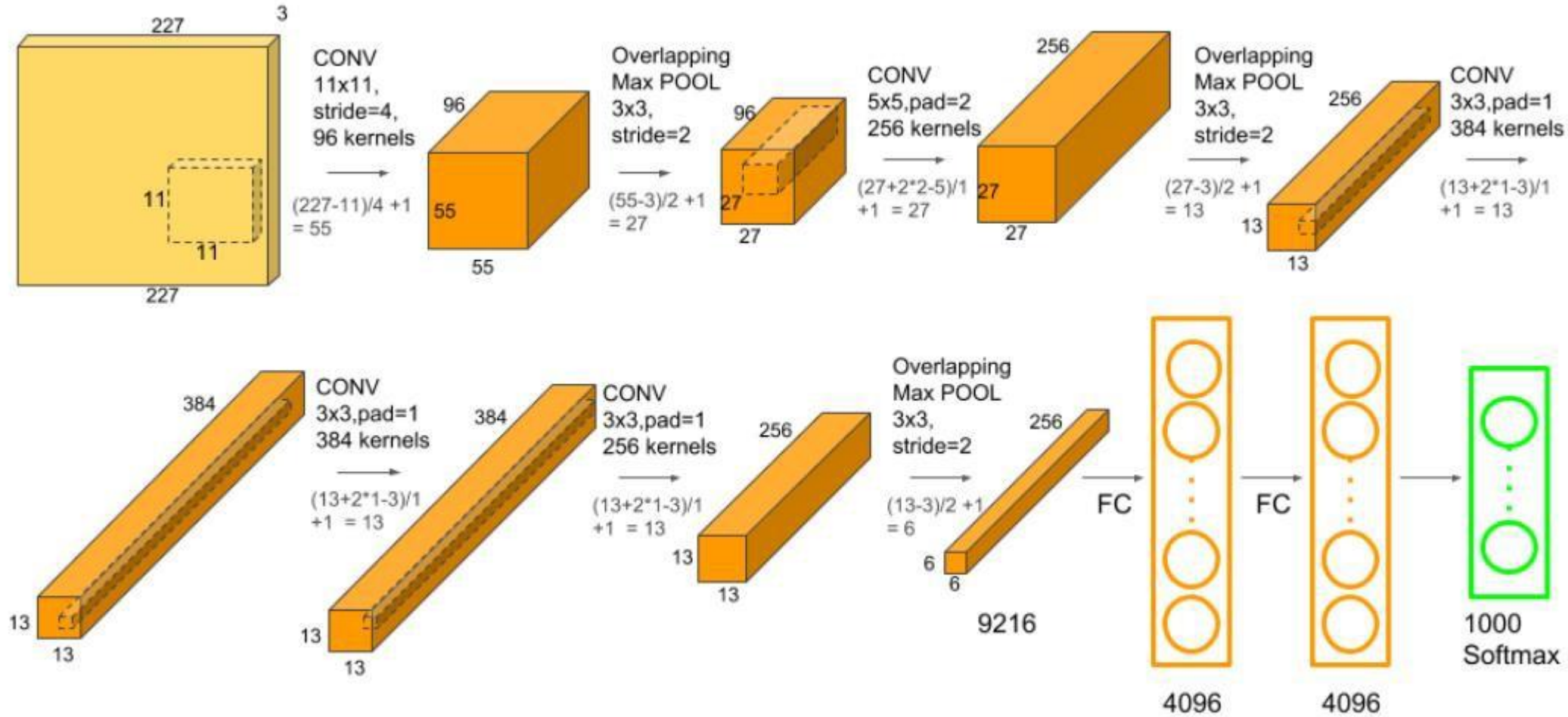
Complex networks have hundreds of millions of weights.

Max Pooling

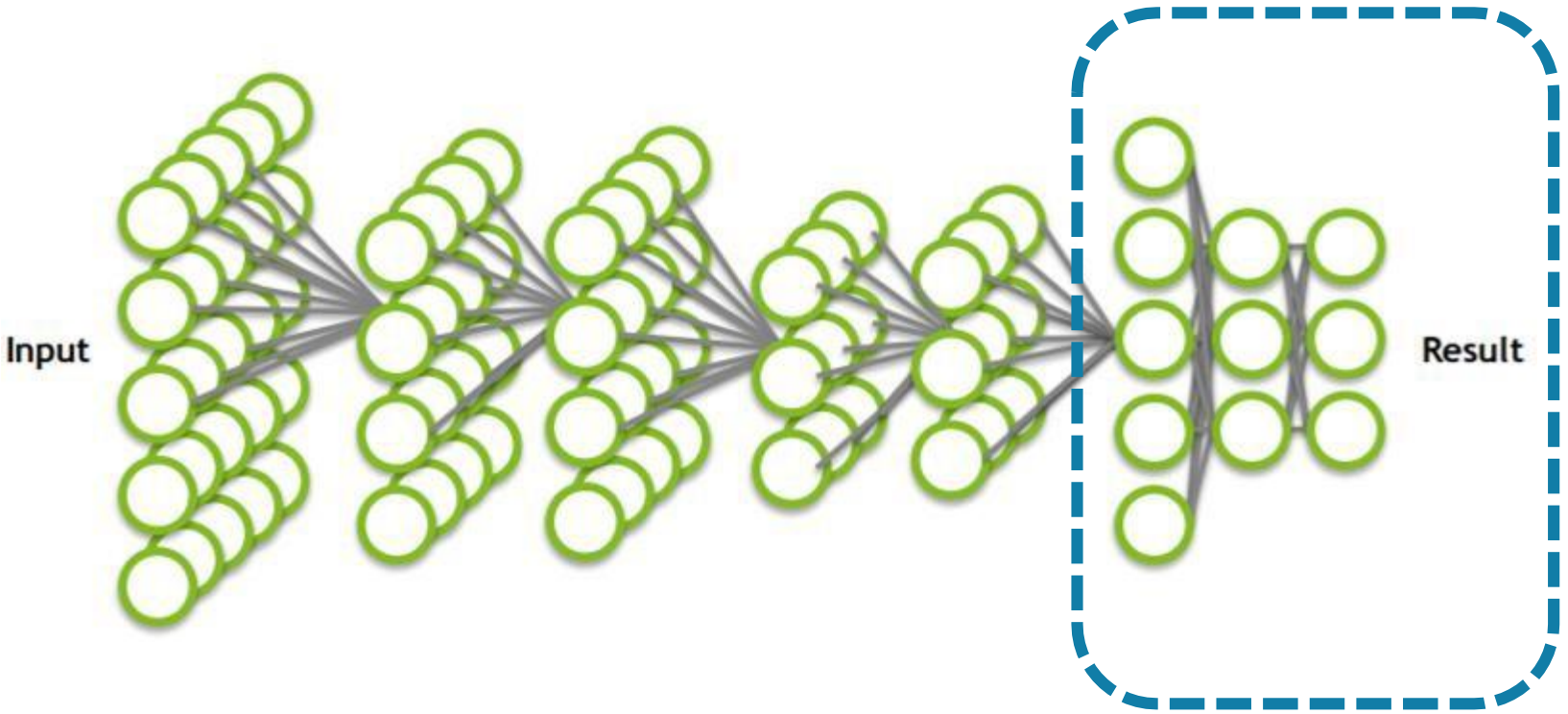


Pooling layer downsamples the volume spatially, independently in each depth slice of the input volume. **Left:** In this example, the input volume of size [224x224x64] is pooled with filter size 2, stride 2 into output volume of size [112x112x64]. Notice that the volume depth is preserved. **Right:** 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).

- Accepts a volume of size $W_1 \times H_1 \times D_1$
- Requires two hyperparameters:
  - their spatial extent F,
  - the stride S,
- Produces a volume of size $W_2 \times H_2 \times D_2$ where:
  - $W_2 = (W_1 - F)/S + 1$
  - $H_2 = (H_1 - F)/S + 1$
  - $D_2 = D_1$
- Introduces zero parameters since it computes a fixed function of the input
- Note that it is not common to use zero-padding for Pooling layers

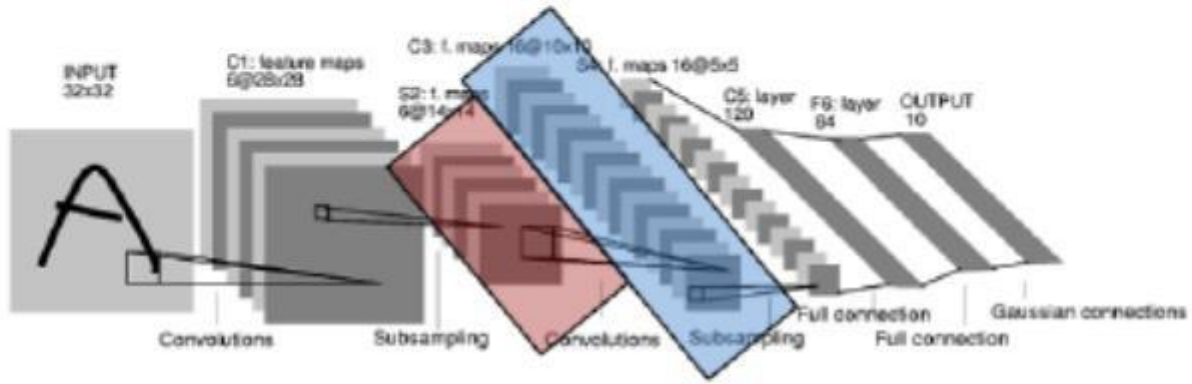

227
3
227
11
11
CONV
11x11,
stride=4,
96 kernels
(227-11)/4 +1
= 55
96
55
55
Overlapping
Max POOL
3x3,
stride=2
(55-3)/2 +1
= 27
96
27
27
CONV
5x5,pad=2
256 kernels
(27+2*2-5)/1
+1 = 27
256
27
27
Overlapping
Max POOL
3x3,
stride=2
(27-3)/2 +1
= 13
256
13
13
CONV
3x3,pad=1
384 kernels
(13+2*1-3)/1
+1 = 13
384
13
13
CONV
3x3,pad=1
384 kernels
(13+2*1-3)/1
+1 = 13
384
13
13
CONV
3x3,pad=1
256 kernels
(13+2*1-3)/1
+1 = 13
256
13
13
Overlapping
Max POOL
3x3,
stride=2
(13-3)/2 +1
= 6
256
6
6
9216
FC
4096
FC
4096
1000
Softmax

FULLY CONNECTED LAYER




```
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dense(50,activation='relu'))
model.add(Dense(6, activation='softmax'))
```

Lots of Parallelism Available in a DNN



- Inputs
- Points of a feature map
- Filters
- Elements within a filter

- Multiplies within layer are independent
- Sums are reductions
- Only layers are dependent
- No data dependent operations
  => can be statically scheduled