CS291K – Advanced Data Mining

Instructor: Xifeng Yan
Computer Science
University of California at Santa Barbara

Convolutional Neural Networks

Lecturer: Fangqiu Han
Computer Science
University of California at Santa Barbara

- The slides are made from:
  - Coursera online course, '**Neural Networks for Machine Learning',** Geoffrey Hinton
  - Coursera online course, '**Machine Learning',** Andrew Ng
  - UCLA summer school for deep learning in 2012
  - Stanford course 'CS231n: Convolutional Neural Networks for Visual Recognition', Fei-Fei Li and Andrej Karpathy
  - Deep Learning – ICML 2013 Tutorial, Yann LeCun

Neural network timeline

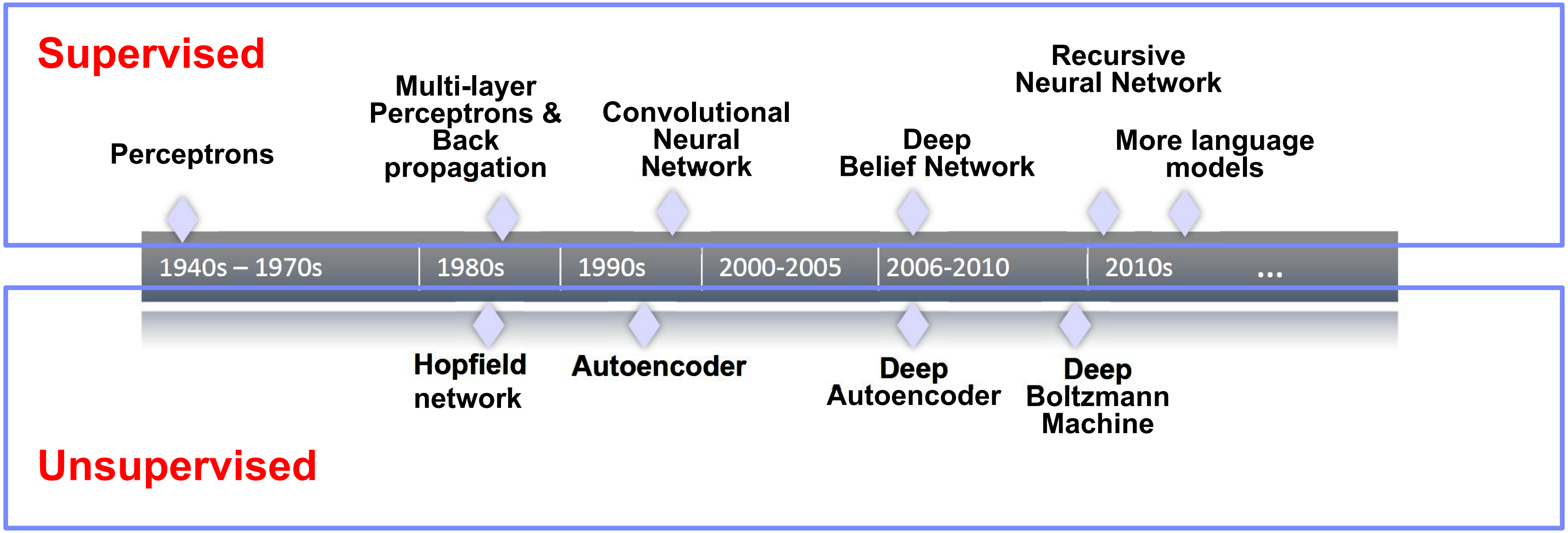


Image Recognition

What do we want computers to do with our data?

- Images/video

Labeling: Motorcycle
Image search

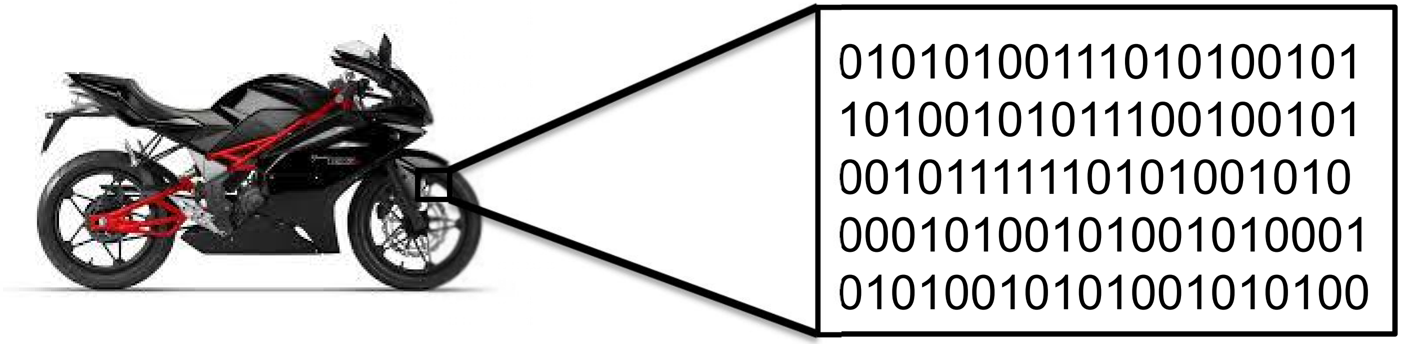
- Audio

Speech recognition
Music classification
Speaker identification

- text

Web search
Anti-spam
Machine translation

Computer vision is hard!



Computer vision is hard!

- Segmentation: Real scenes are cluttered with other objects

Computer vision is hard!

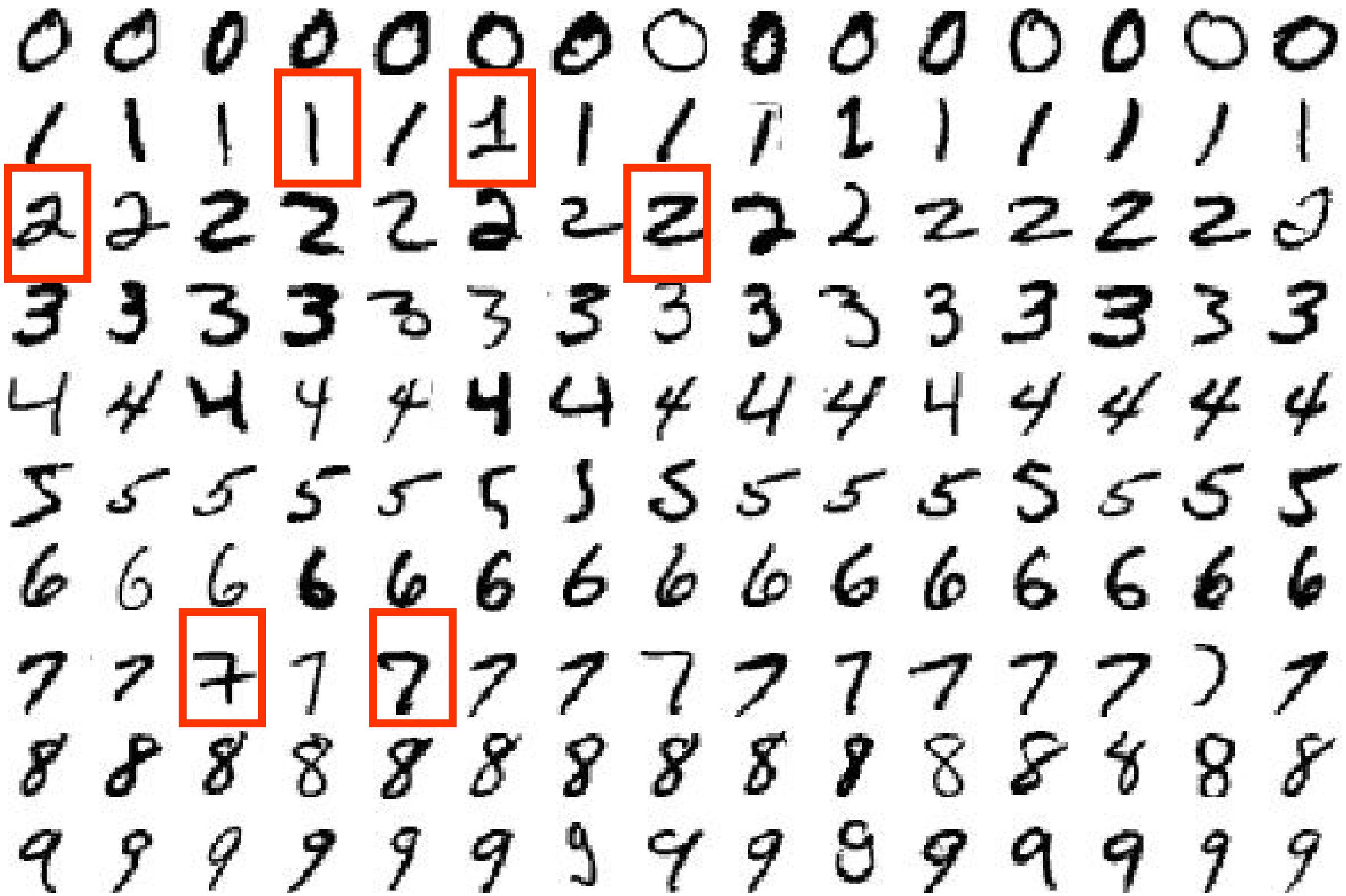
- Segmentation: Real scenes are cluttered with other objects.
- Lighting: The intensities of the pixels are determined as much by the lighting as by the objects.

Computer vision is hard!

- Segmentation: Real scenes are cluttered with other objects.
- Lighting: The intensities of the pixels are determined as much by the lighting as by the objects.
- Deformation: Objects can deform in a variety of non-affine ways

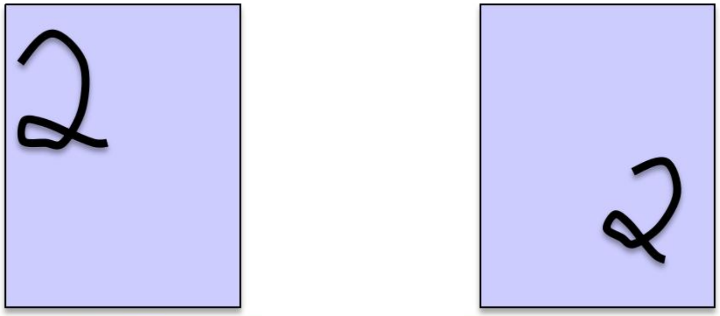


Deformation



Computer vision is hard!

- Segmentation: Real scenes are cluttered with other objects.
- Lighting: The intensities of the pixels are determined as much by the lighting as by the objects.
- Deformation: Objects can deform in a variety of non-affine ways
- Viewpoint: Changes in viewpoint cause changes in images that standard learning methods cannot cope with.

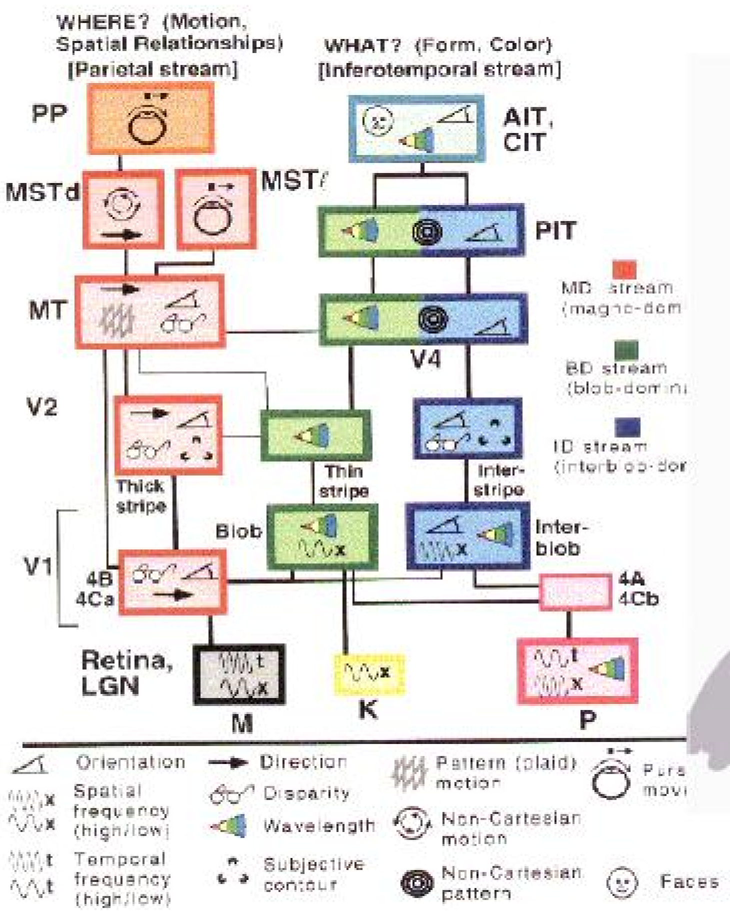


Computer vision is hard!

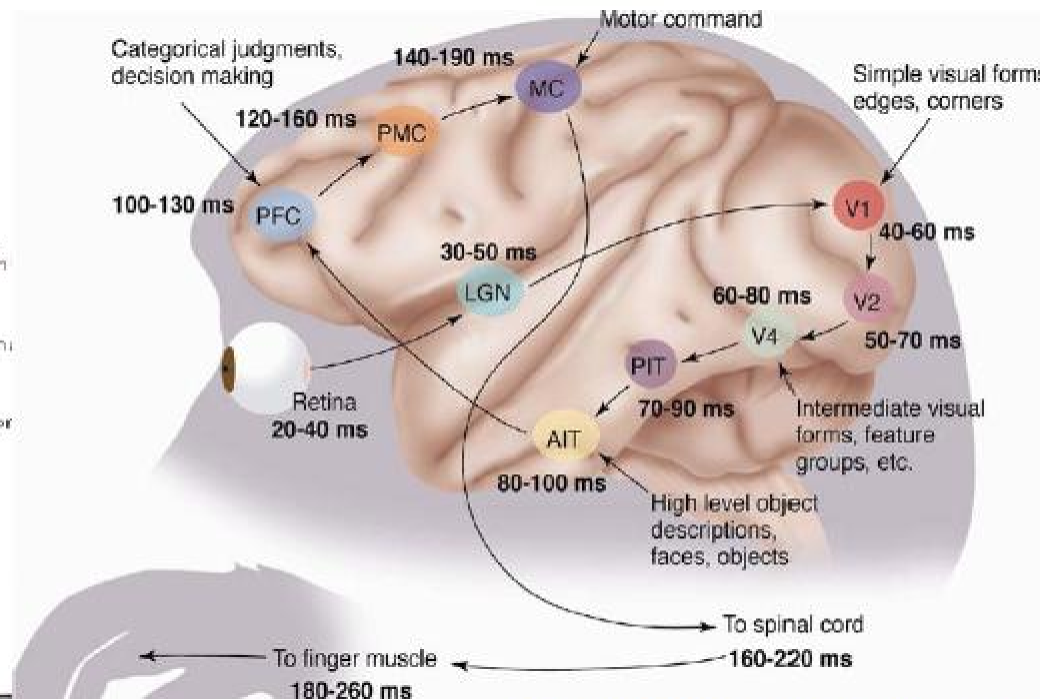
- Segmentation: Real scenes are cluttered with other objects.
- Lighting: The intensities of the pixels are determined as much by the lighting as by the objects.
- Deformation: Objects can deform in a variety of non-affine ways
- Viewpoint: Changes in viewpoint cause changes in images that standard learning methods cannot cope with.

The Mammalian Visual Cortex is Hierarchical

- The ventral (recognition) pathway in the visual cortex has multiple stages: Retina - LGN - V1 - V2 - V4 - PIT - AIT ....



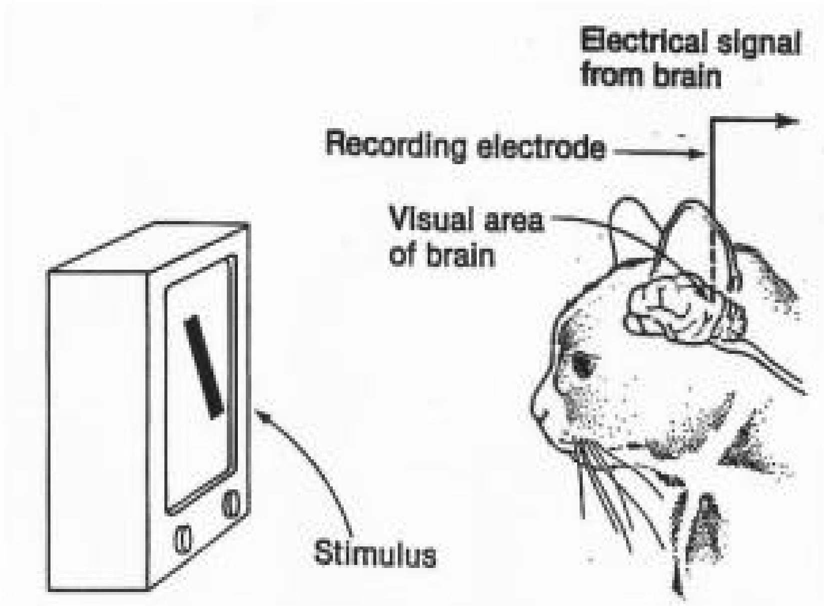
[Gallant & Van Essen]



[picture from Simon Thorpe]

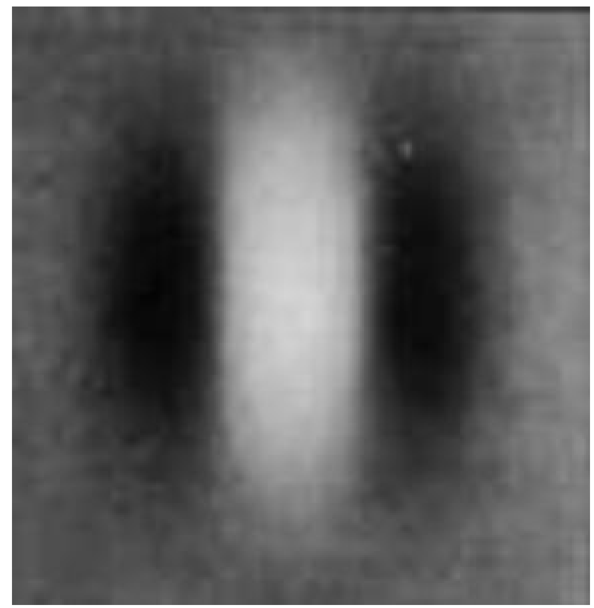
First stage of visual processing: V1

- Hubel & Wiesel, 1959, receptive fields of single neuron in the cat`s visual cortex



First stage of visual processing: V1

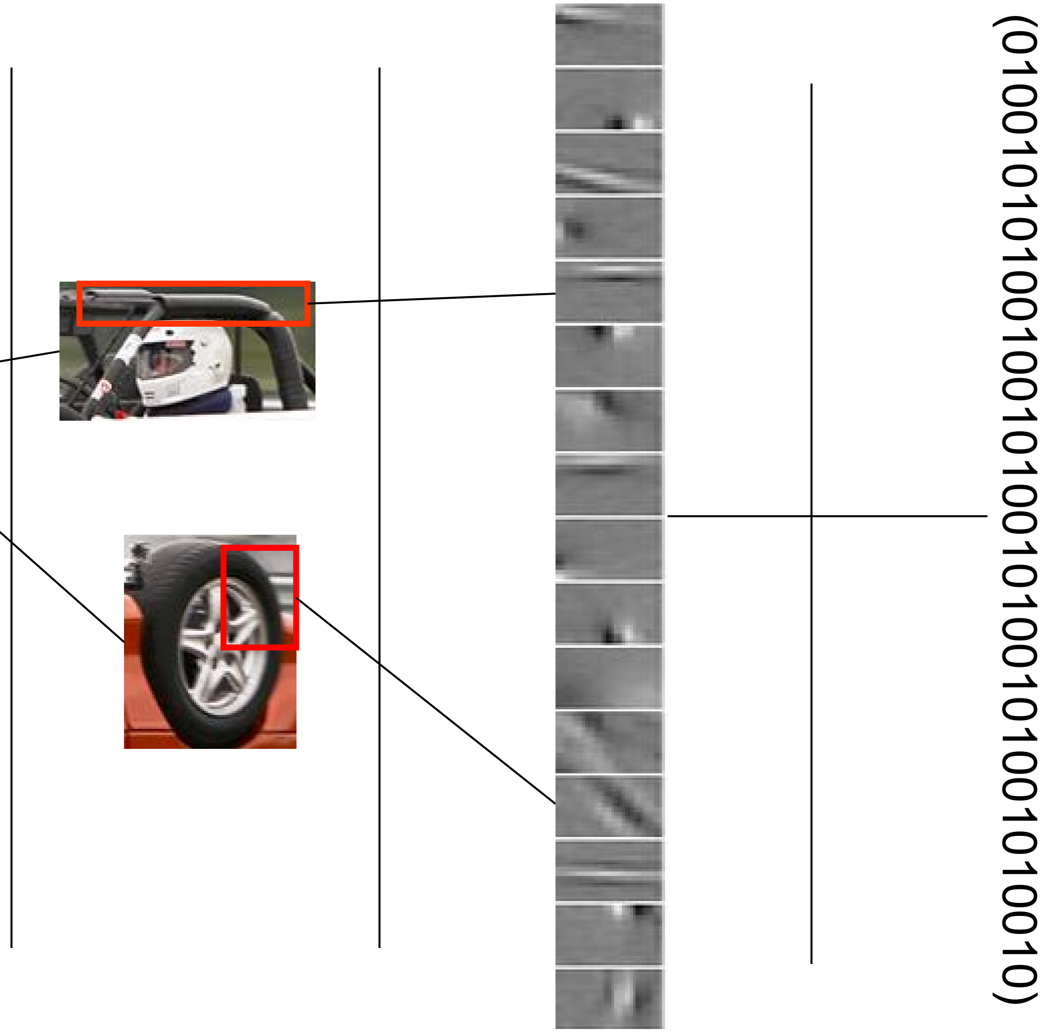
- Hubel & Wiesel, 1959, receptive fields of single neuron in the cat`s visual cortex



Neuron #1 of visual cortex

Neuron #2 of visual cortex

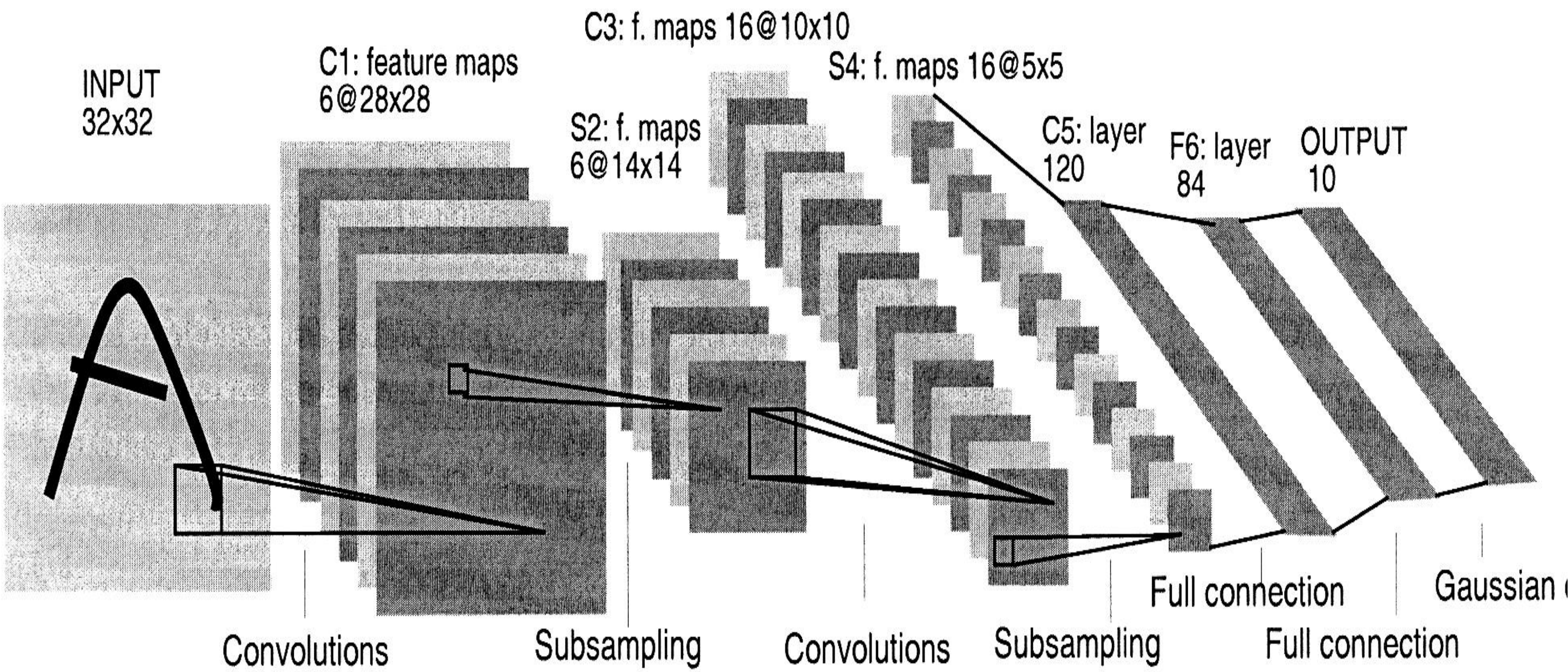
Why deep learning – Recognizing deep features



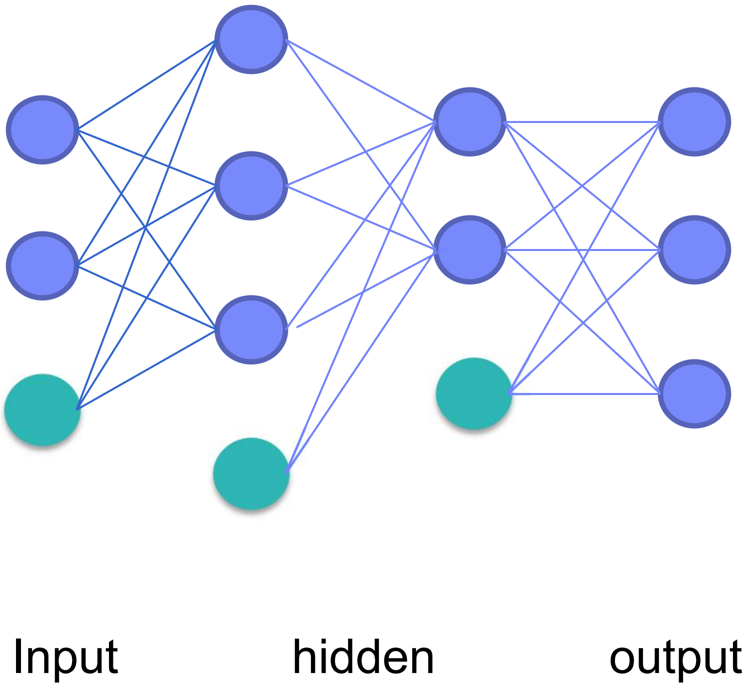
- ❑ Object Recognition


CONVNETS
HOW DO THEY WORK

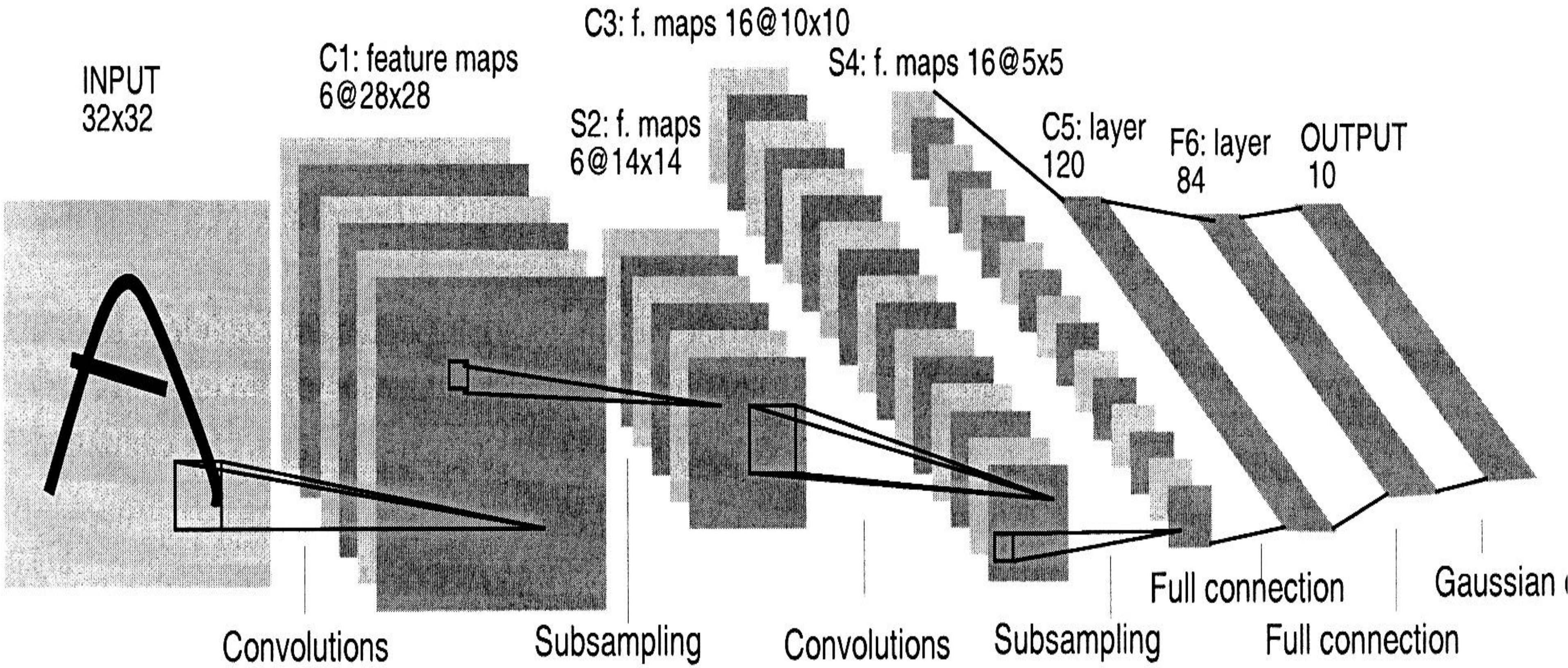
The architecture of LeNet5



Multi-layer Perceptrons



The architecture of LeNet5



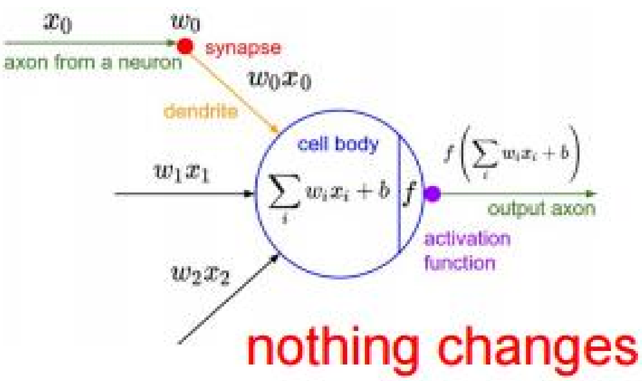
Outline

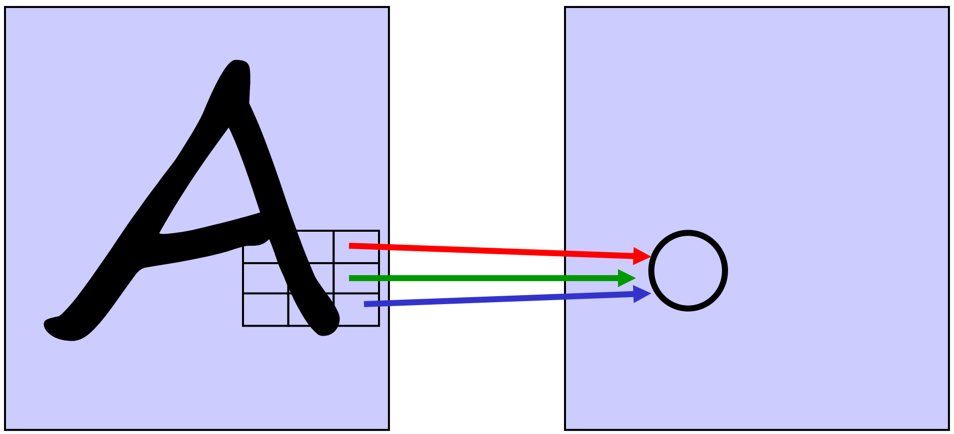
- Local connectivity
- Replicated feature (weight sharing)
- Subsampling (pooling)

Convolutional Layer

Subsampling Layer

Local connectivity

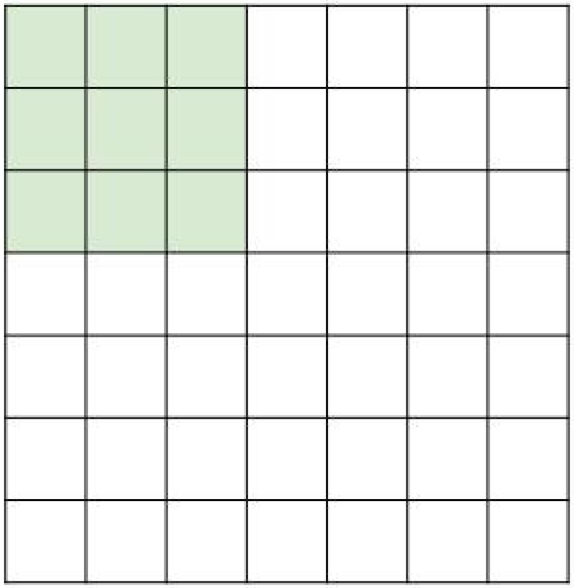




Input Layer
32 × 32

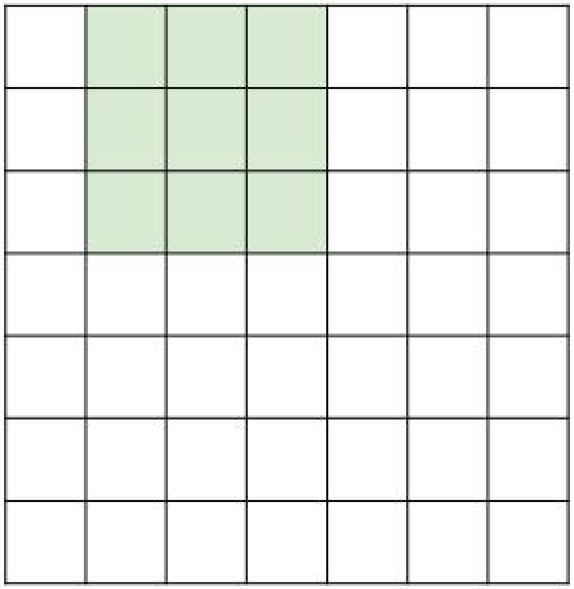
Convolutional Layer 1
30 × 30

First neuron in the convolutional layer



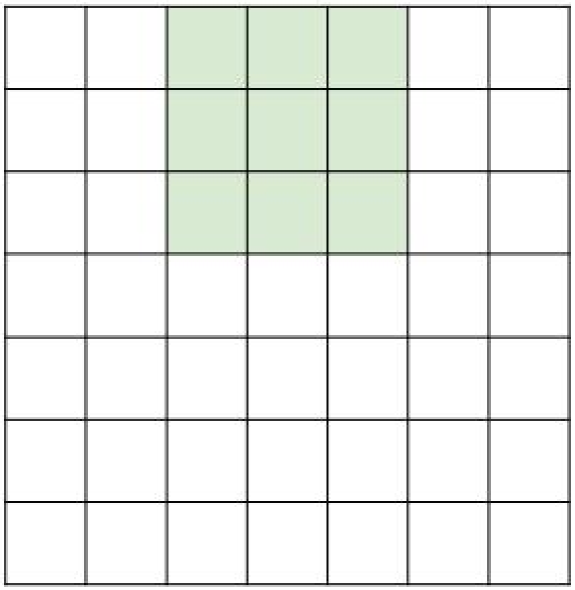
7 × 7 input image

Second neuron in the convolutional layer



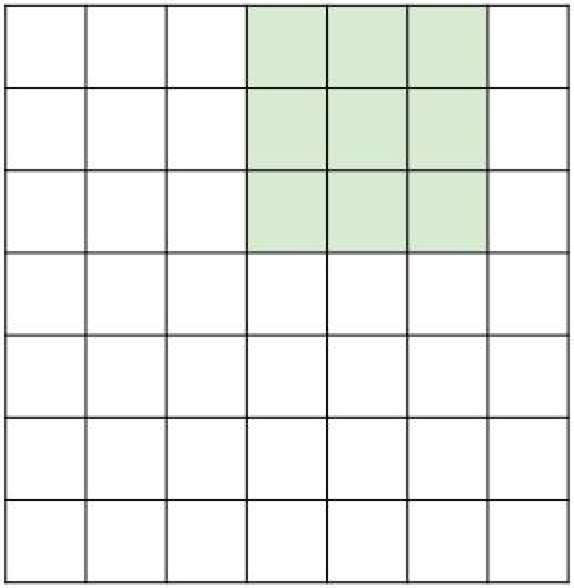
7 × 7 input image

Third neuron in the convolutional layer



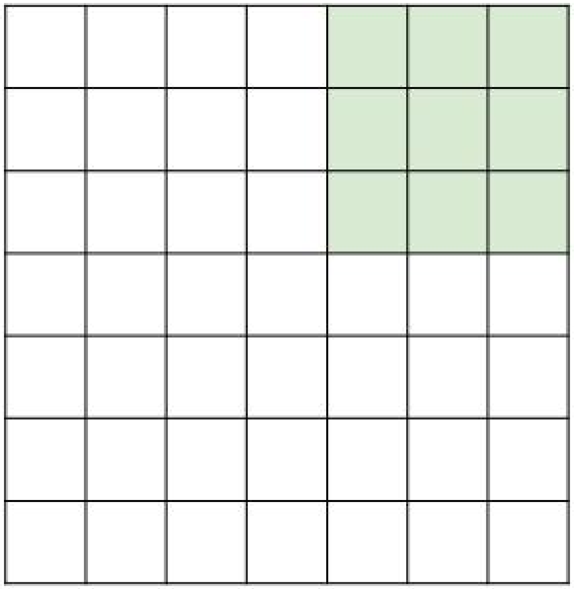
7 × 7 input image

Fourth neuron in the convolutional layer



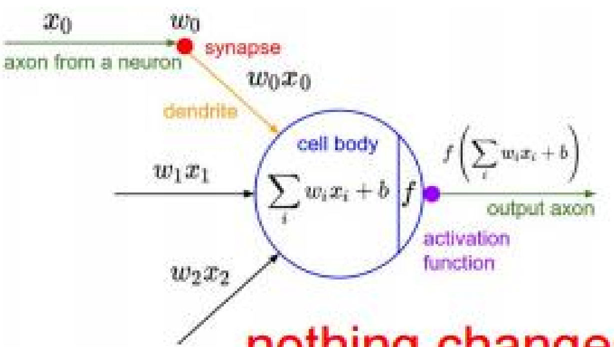
7 × 7 input image

Fifth neuron in the convolutional layer

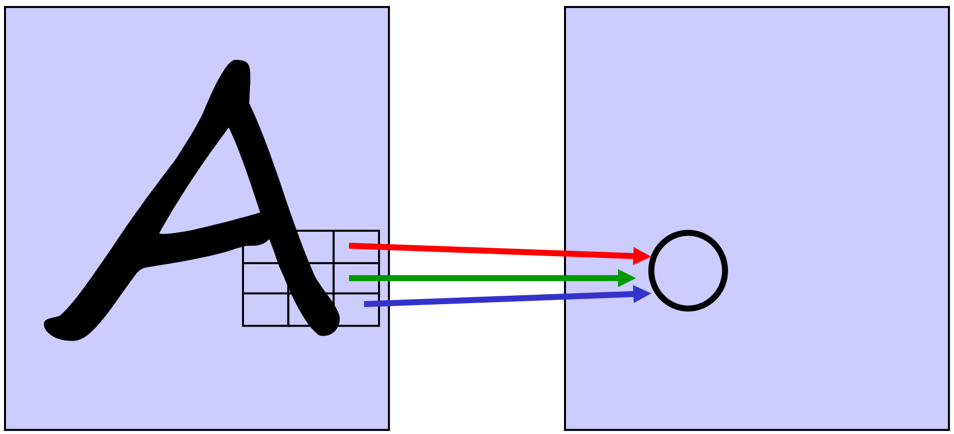


7 × 7 input image

Local connectivity



nothing changes

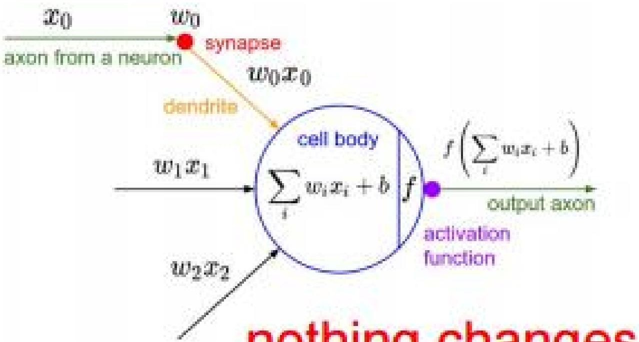


Input Layer
32 × 32

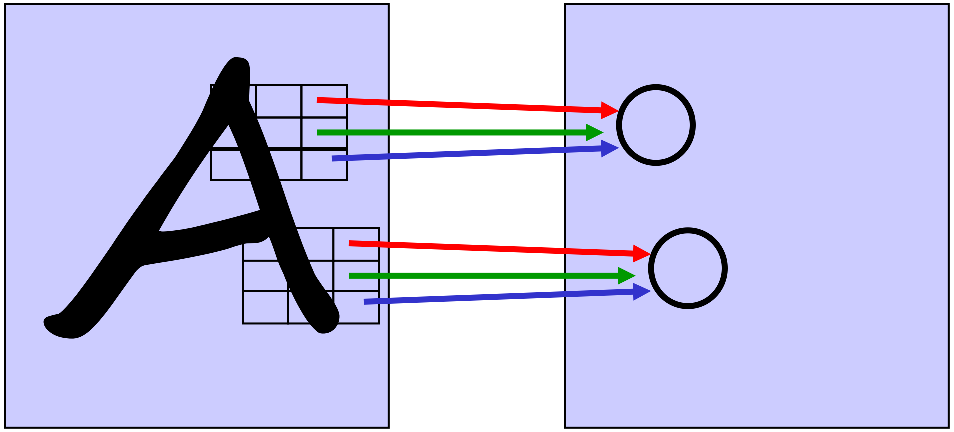
Convolutional Layer 1
30 × 30

Total parameter number: 9 × 30 × 30

Weight sharing



nothing changes



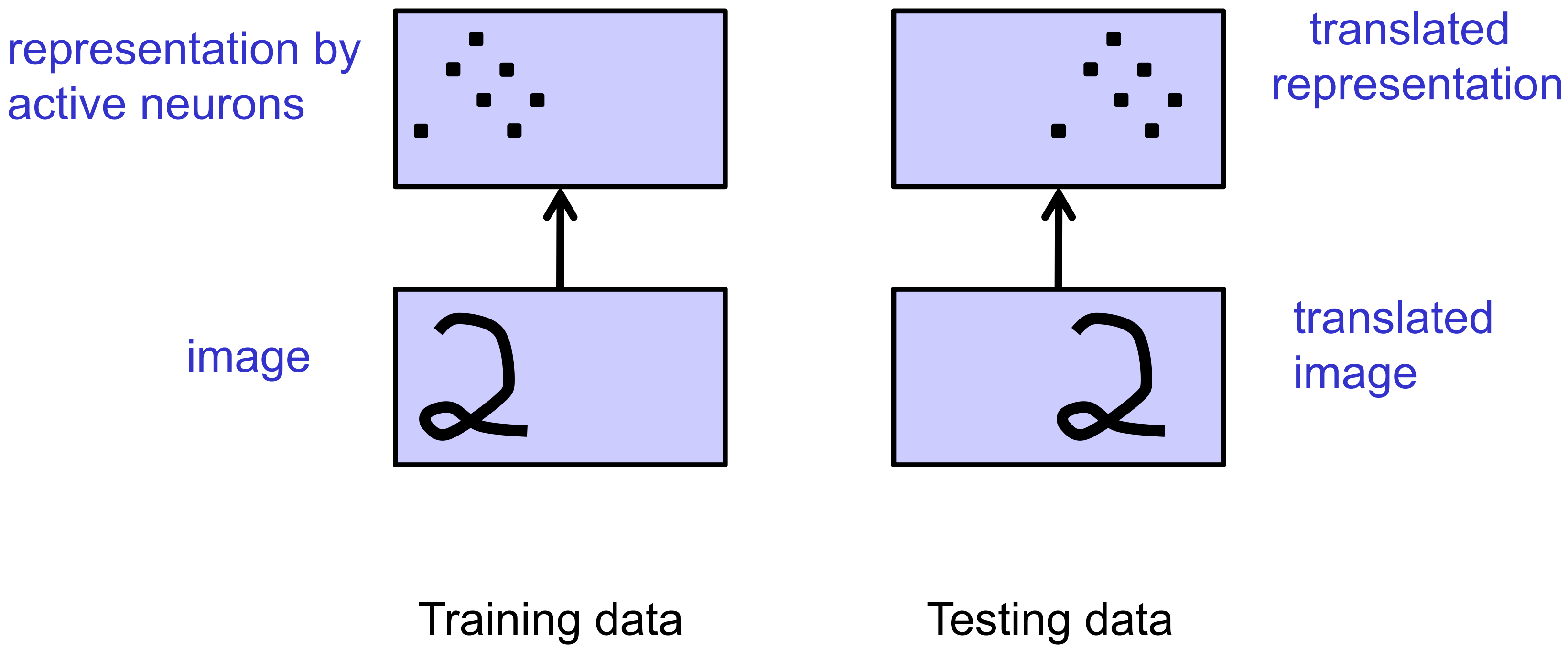
Input Layer
32 × 32

Convolutional Layer 1
30 × 30

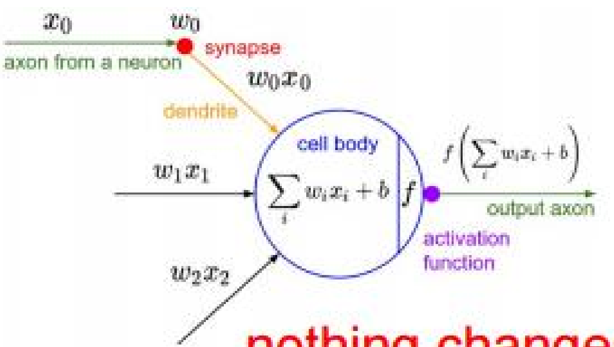
Total parameter number: 9

What does replicating the feature detectors achieve?

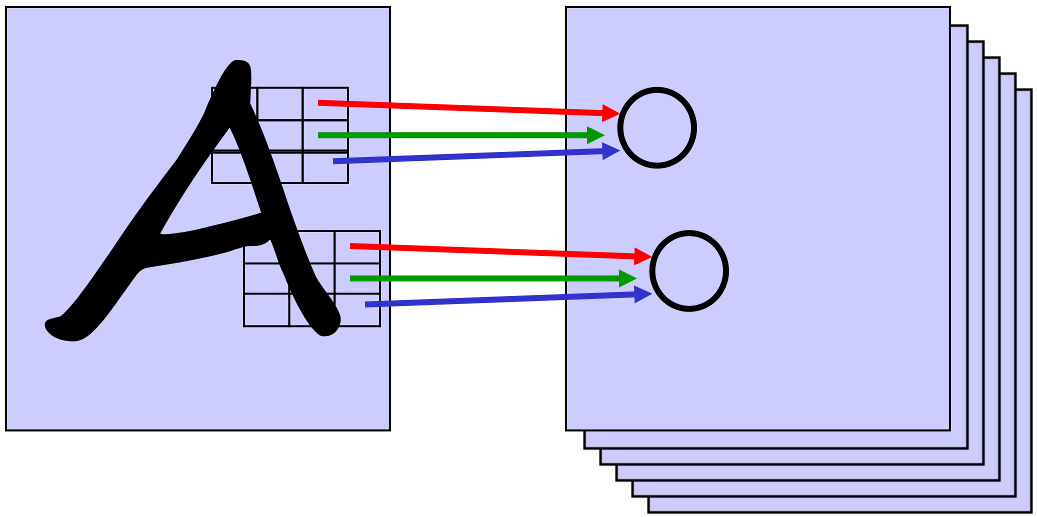
- Invariant knowledge: If a feature is useful in some locations during training, detectors for that feature will be available in all locations during testing.



Multiple feature maps



nothing changes



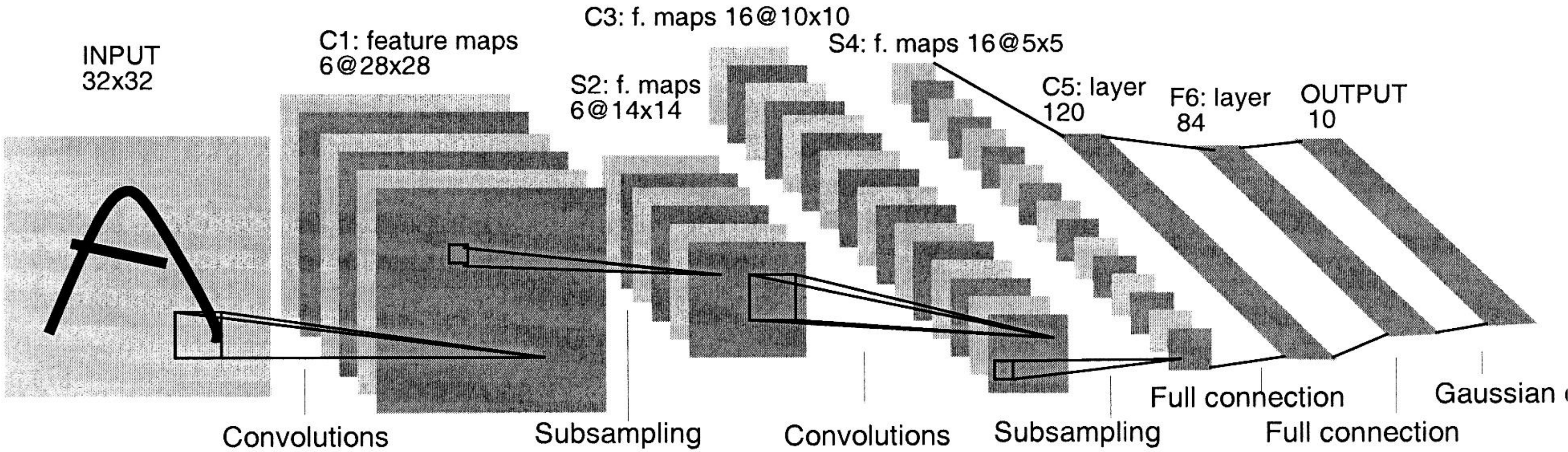
Multiple neurons all looking at the same region of the input.

Input Layer
32 × 32

Convolutional Layer 1
30 × 30

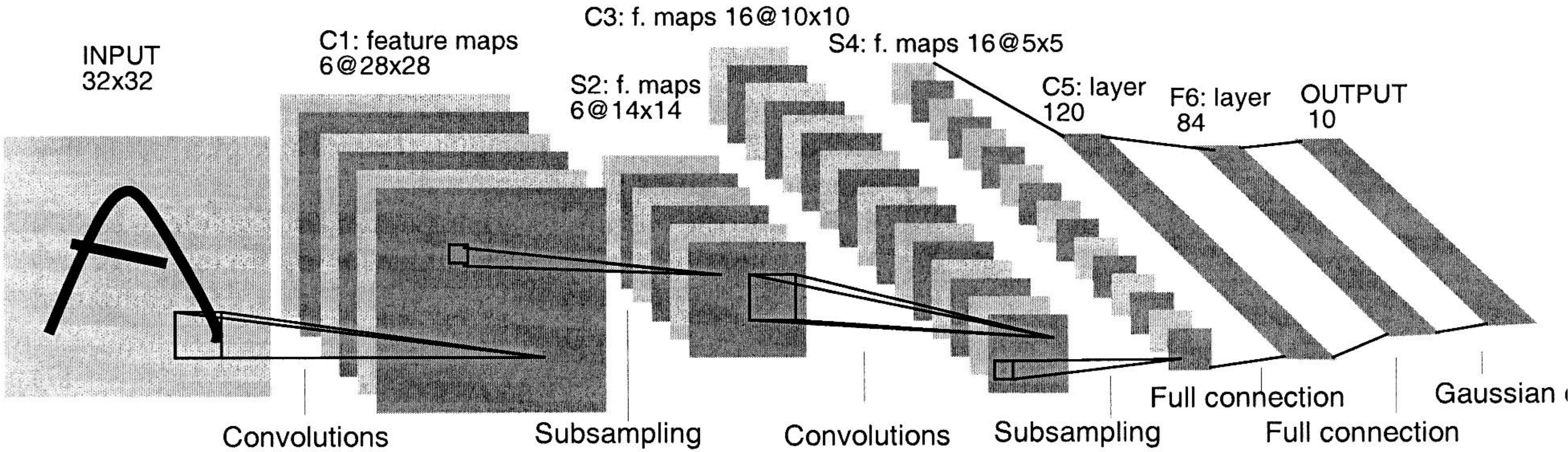
Total parameter number: 9 × 6

LeNet 5, Layer C1



- C1: Convolutional layer with 6 feature maps of size 28x28.
- Each unit of C1 has a 5x5 receptive field in the input layer.
- Total number of parameters: (5*5+1)*6=156.
- Total connections: (32*32+1)*(28*28)*6.

LeNet 5, Layer C3



- C3: Convolutional layer with 16 feature maps of size 10x10.
- Each unit in C3 is connected to **several** 5x5 receptive fields at identical locations in S2 Local connections.
- Total number of parameters: 1516.

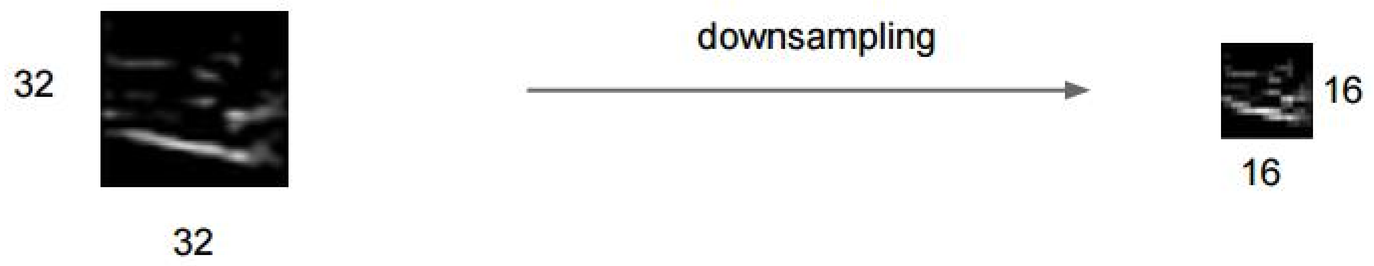
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	X				X	X	X			X	X	X	X		X	X
1	X	X				X	X	X			X	X	X	X		X
2	X	X	X				X	X	X			X		X	X	X
3		X	X	X			X	X	X	X			X		X	X
4			X	X	X			X	X	X	X		X	X		X
5				X	X	X			X	X	X	X		X	X	X

TABLE I

EACH COLUMN INDICATES WHICH FEATURE MAP IN S2 ARE COMBINED BY THE UNITS IN A PARTICULAR FEATURE MAP OF C3.

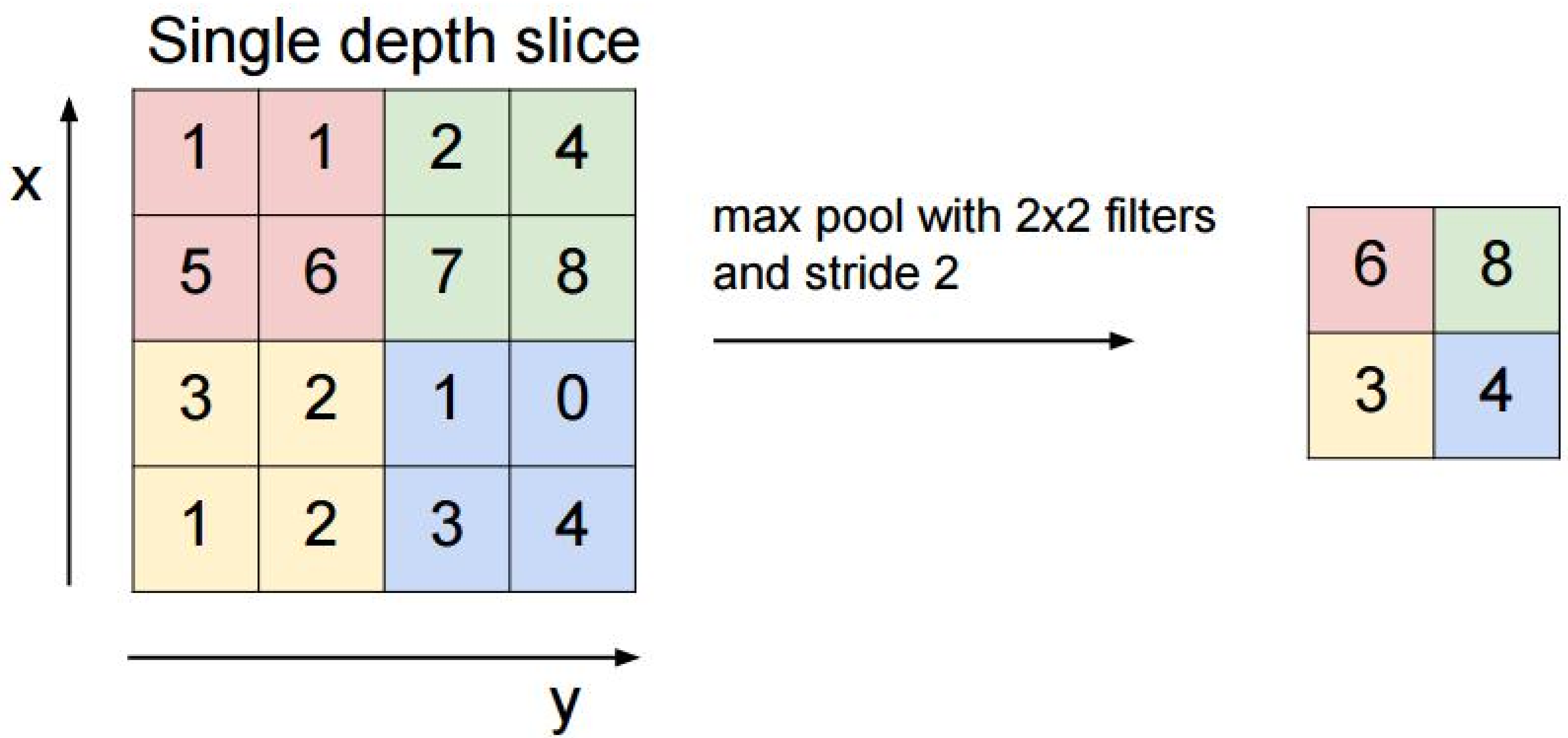
Pooling

- Makes the representations smaller.



Pooling

- Makes the representations smaller.
- Aggregating four neighboring activations to give a single output to the next level.
  - Average, Max, Sum, Lp norm etc.



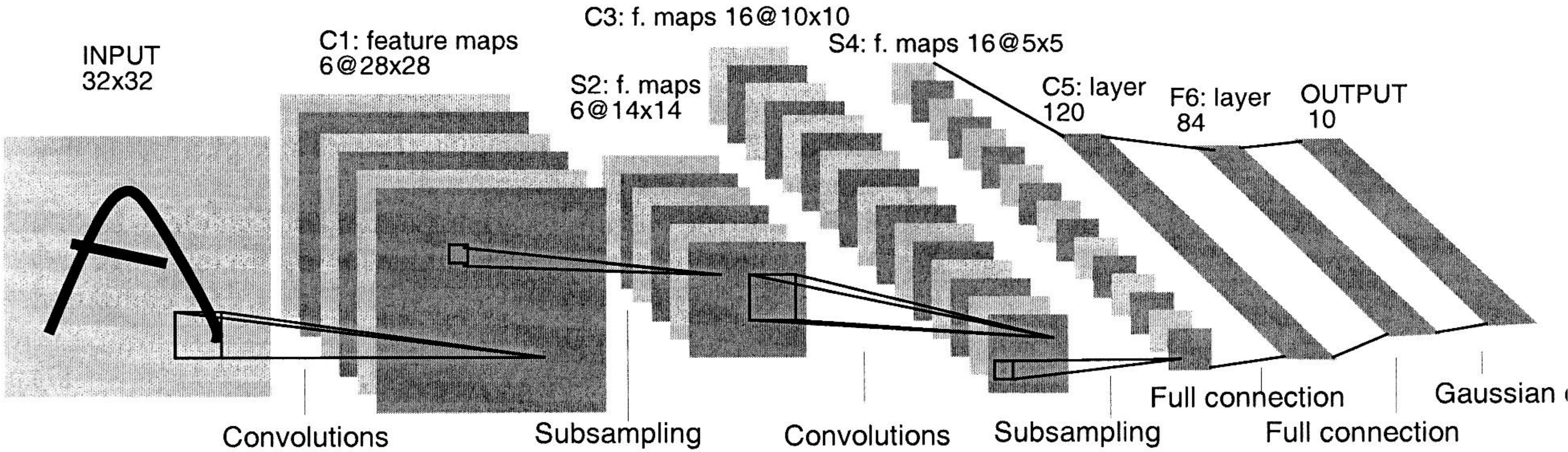
Why pooling

- A feature (of the right size) usually does not appear twice in a small neighborhood.

- Reduces the number of inputs to the next layer of feature extraction, thus allowing us to have many more different feature maps.

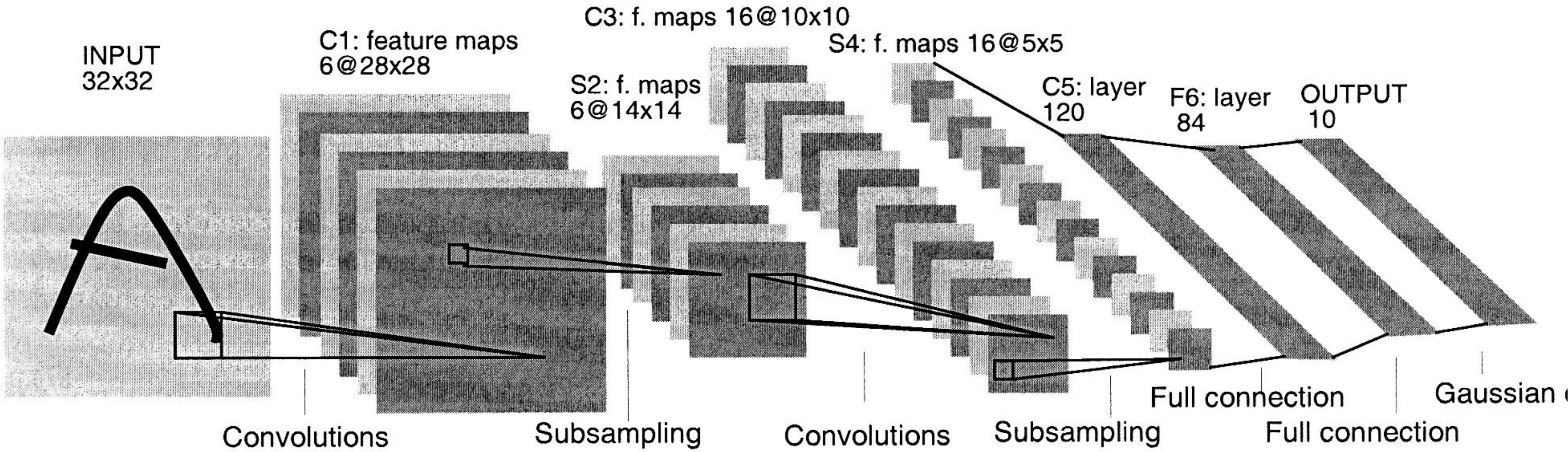
- Get a small amount of translational invariance at each level.

LeNet 5, Layer S2



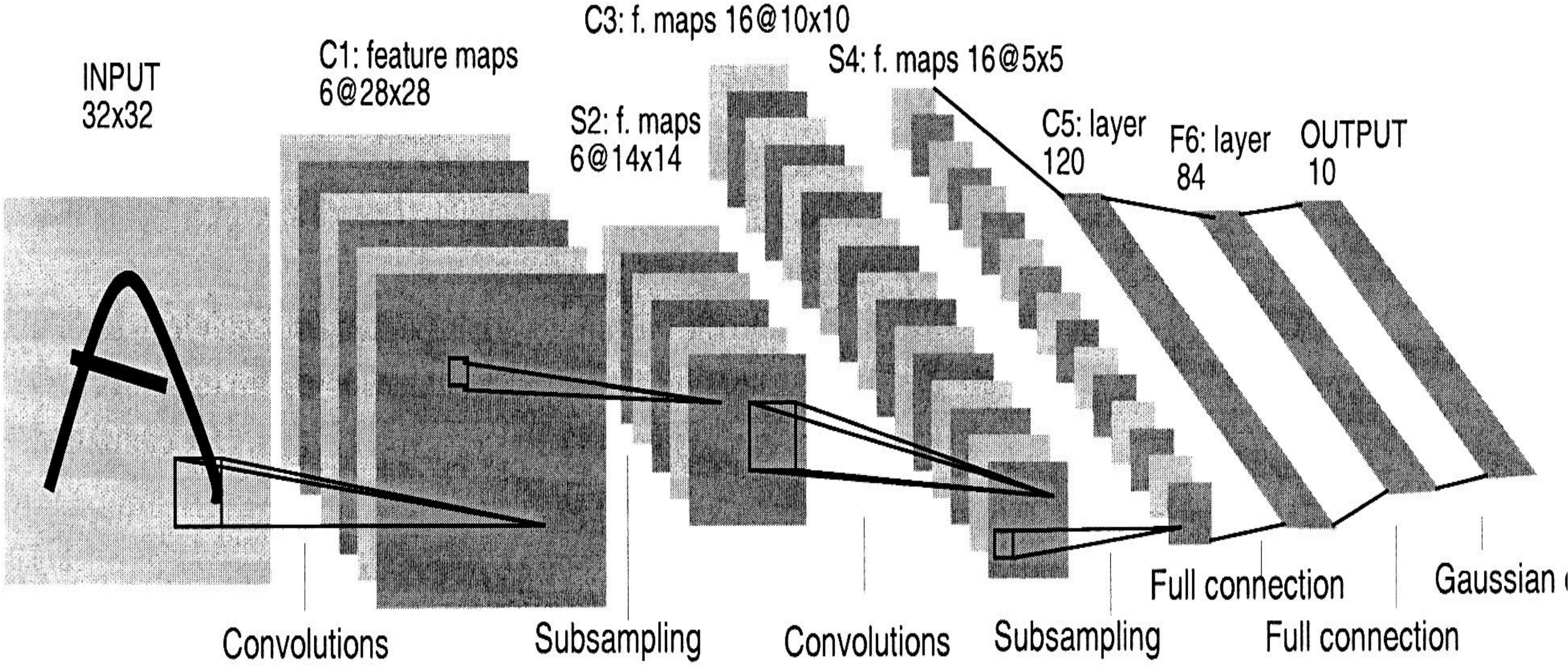
- S2: Subsampling layer with 6 feature maps of size 14 x 14
- 2x2 nonoverlapping receptive fields in C1
- Total number of parameters: 0

LeNet 5, Layer S4



- S4: Subsampling layer with 6 feature maps of size 5 x 5
- 2x2 nonoverlapping receptive fields in C3
- Total number of parameters: 0

The architecture of LeNet5



LeNet 5 Training

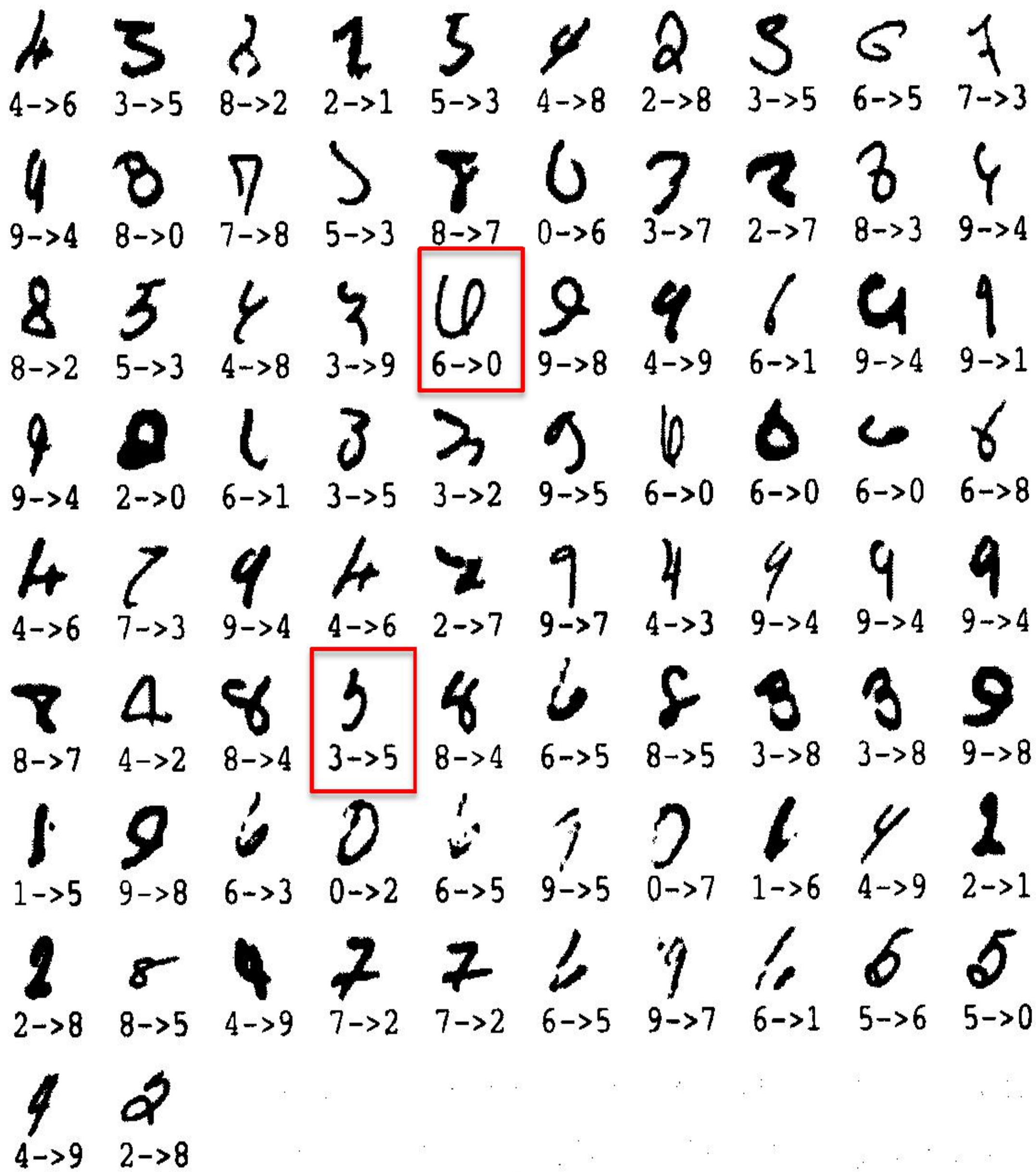
- Backpropagation algorithm with constrain.

- To constrain $W_1 = W_2$
  - We need same initialization.
  - We need $\Delta W_1 = \Delta W_2$

- Use $\frac{\partial E}{\partial w_1} + \frac{\partial E}{\partial w_2}$ for both W_1 and W_2 .

MNIST Dataset

- Original datasets:
  - 60,000 handwritten digits for training
  - 10,000 for testing
- [Dateset website]()



The 82 errors made by LeNet5

The human error rate is probably 20 to 30 errors but nobody has had the patience to measure it.

Demo

Priors

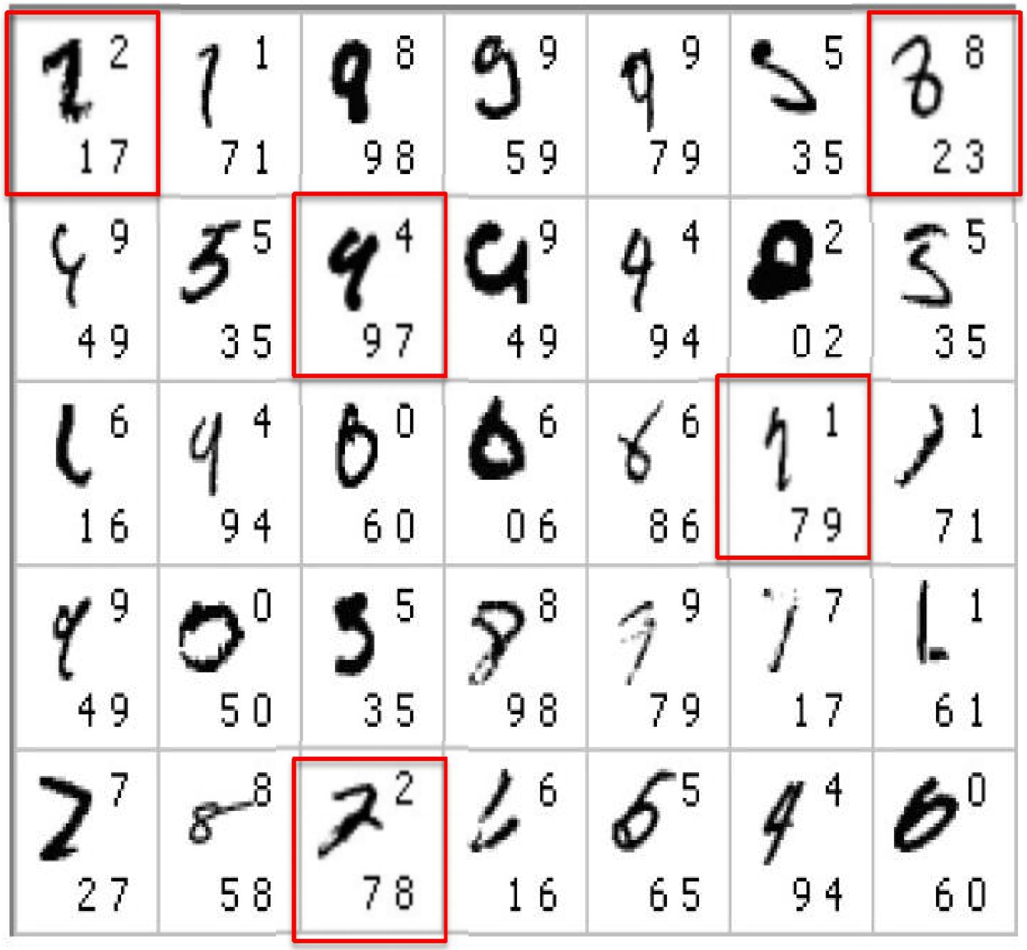
- We can put our prior knowledge about the task into the network by designing appropriate:
  - Local connectivity
  - Weight sharing
  - Neuron activation functions

- Alternatively, we can use our prior knowledge to create a whole lot more training data.
  - For each training image, produce many new training examples by applying many different transformations.

MNIST Dataset

- Original datasets:
  - 60,000 handwritten digits for training
  - 10,000 for testing
- Distorting datasets:
  - Using shifts, scaling, skewing, and compression
  - 540,000 + 60,000 handwritten digits
- Dateset website

The errors made by the Ciresan *et. al.* net



The top printed digit is the right answer. The bottom two printed digits are the network's best two guesses.

- Structure: 1-20-P-40-P-150-10
- The right answer is almost always in the top 2 guesses.
- With model averaging they can now get about 25 errors.
- Best results on MNIST

Recommendation readings / videos

- Coursera:
  - Neural Networks for Machine Learning, Geoffrey Hinton
  - Machine Learninig, Andrew Ng
- Tutorial:
  - Neural Networks and Deep Learning: http://neuralnetworksanddeeplearning.com/
  - http://deeplearning.net/tutorial
  - UCLA deep learning summer school
  - A tutorial on Deep Learning – NIPS 2009 Tutorial, Geoffrey Hinton
  - Representation Learning Tutorial – ICML 2012 Tutorial, Yoshua Bengio
  - Deep Learning – ICML 2013 Tutorial, Yann LeCun

Questions?