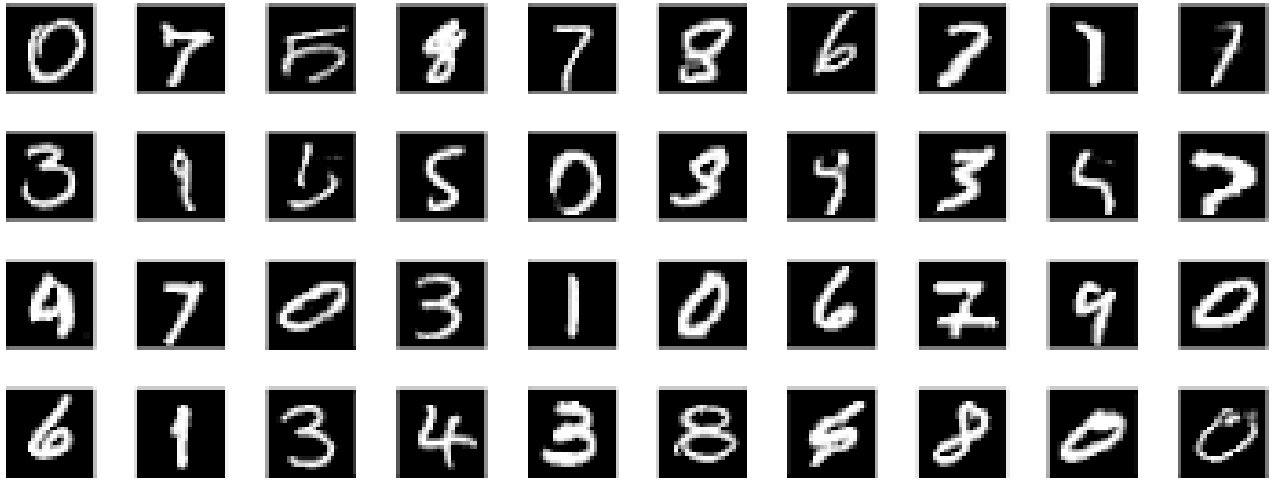
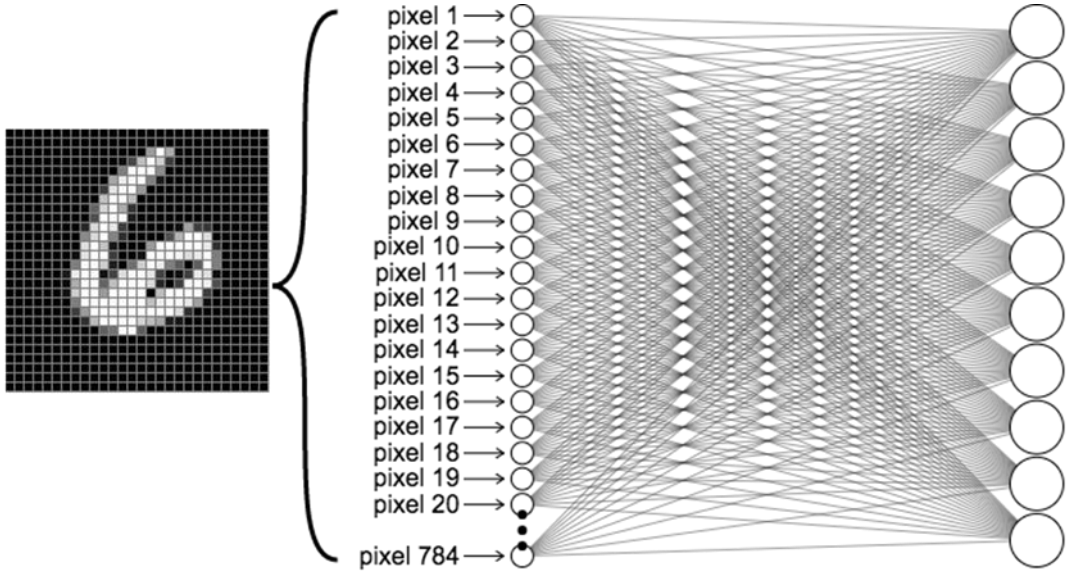
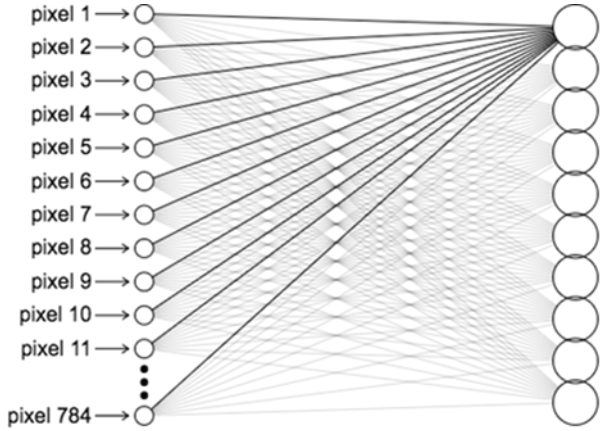
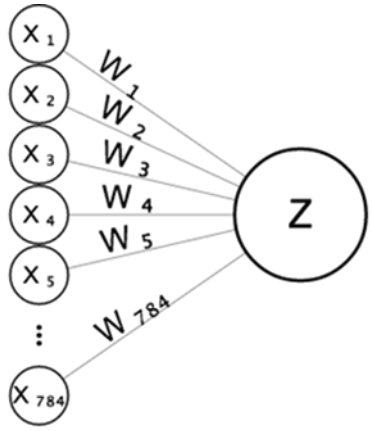




While solving an image classification problem using ANN, the first step is to convert a 2-dimensional image into a 1-dimensional vector prior to training the model. This has two drawbacks:

- The number of trainable parameters increases drastically with an increase in the size of the image.
- The spatial relationship between the pixels are not considered in ANN.





Scalability is an issue with ANN

David Hubel and Torsten Wiesel

1981 Nobel Prize in Physiology or Medicine for their discoveries concerning information processing in the visual system

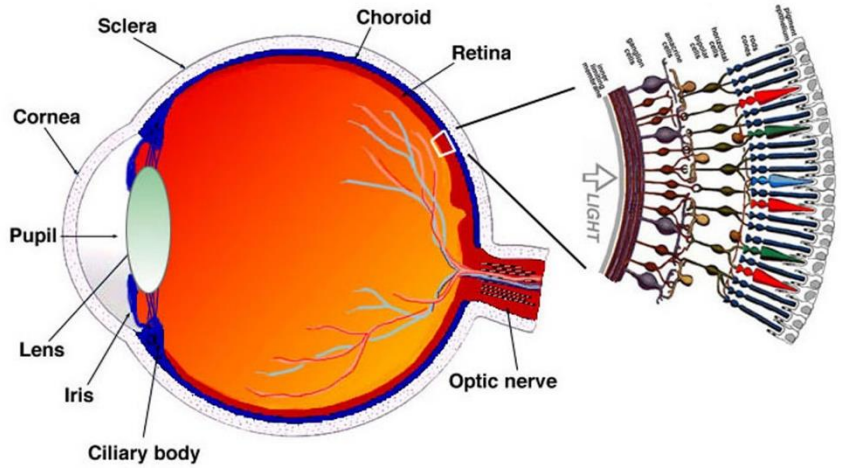
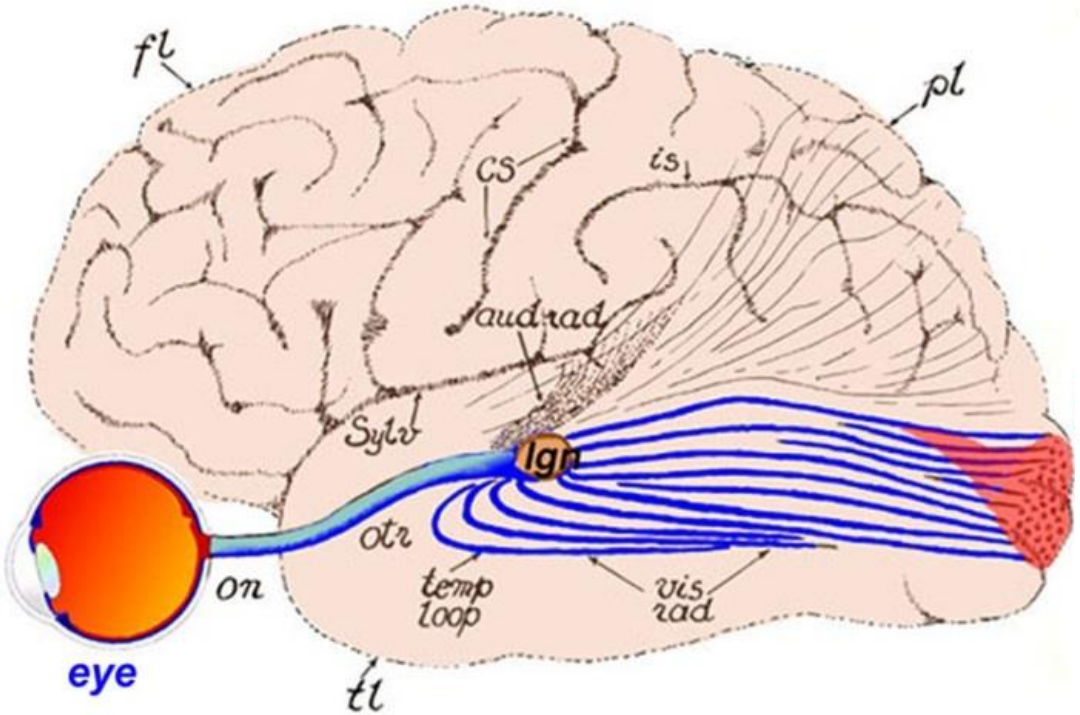
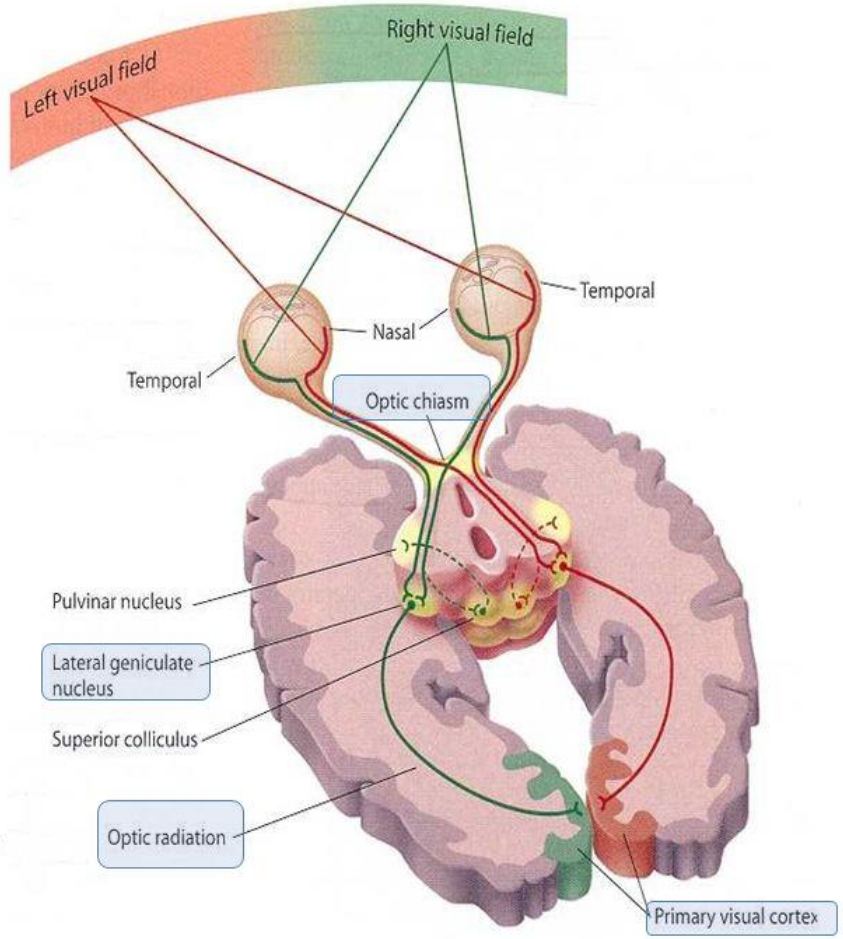


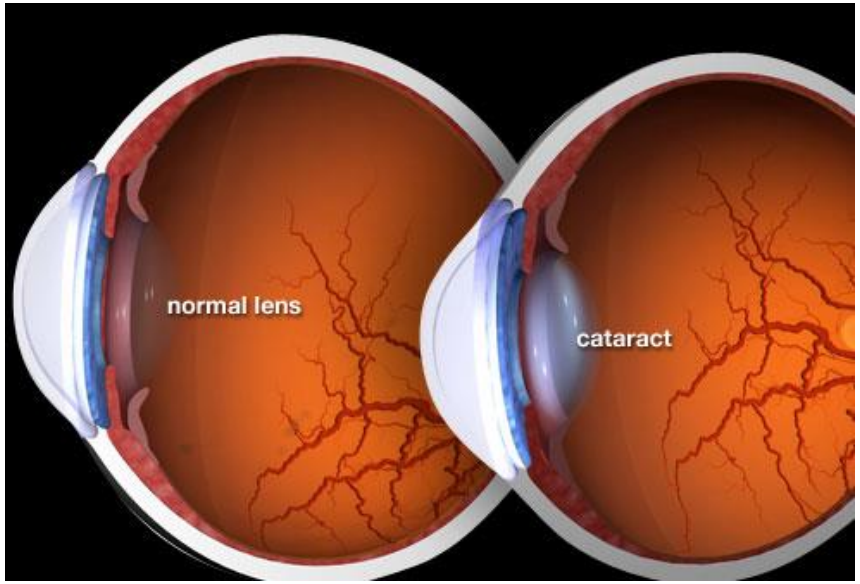
Fig. 1.1. A drawing of a section through the human eye with a schematic enlargement of the retina.

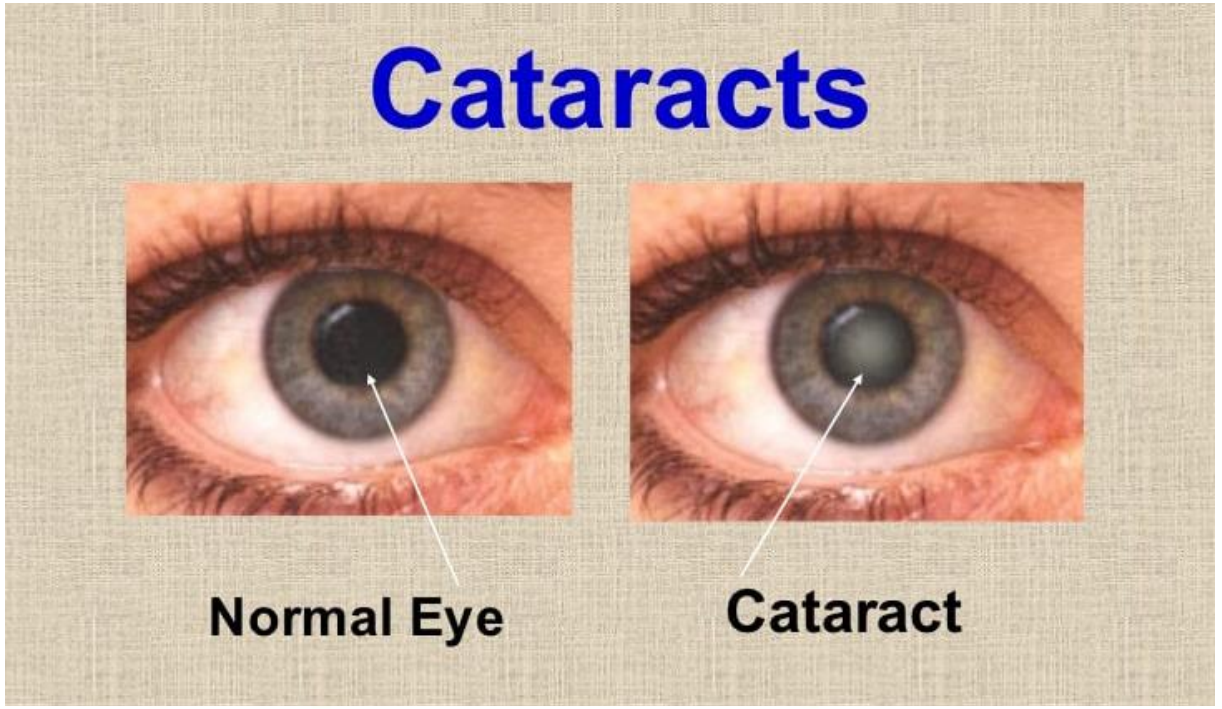


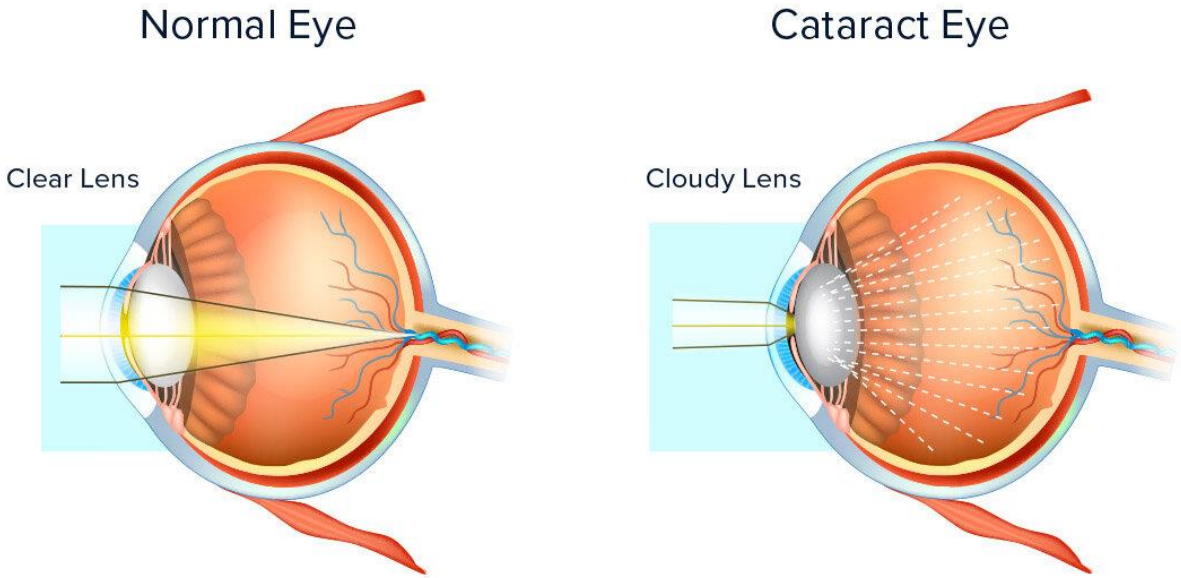
Visual Pathway

1. Cones
2. Bipolar neurons
3. Ganglion cell's axon forms the optic nerve
4. Optic nerve to the Optic Chiasm
5. Optic tract
6. Lateral geniculate nuclei of the thalamus
7. Optic Radiations
8. Primary visual areas of the occipital lobes

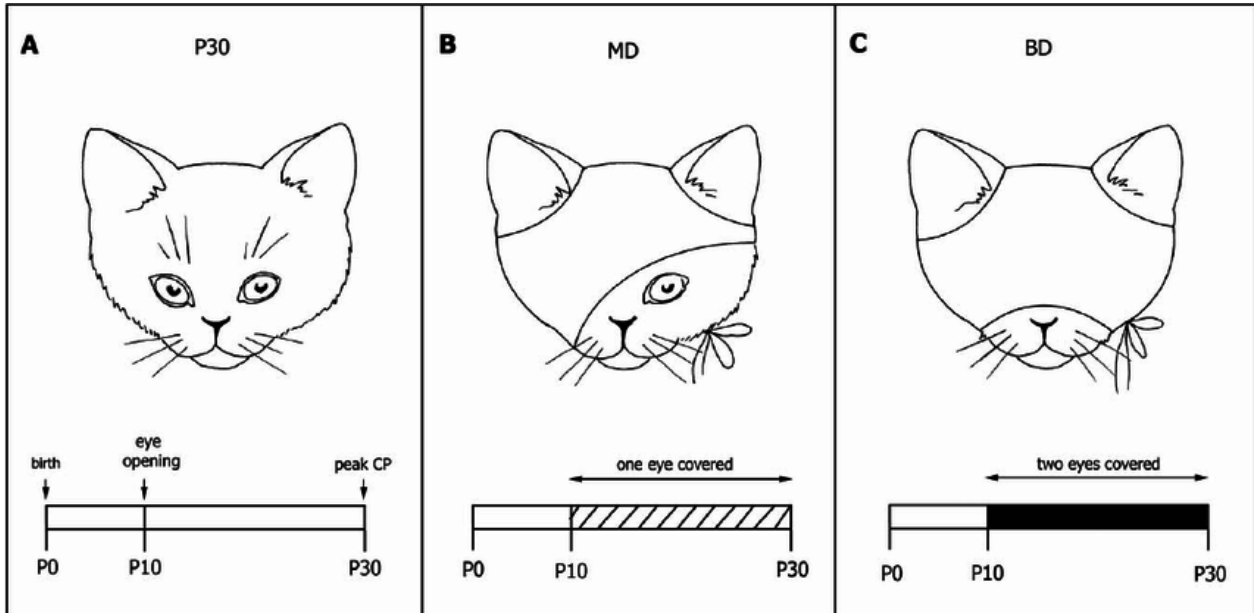


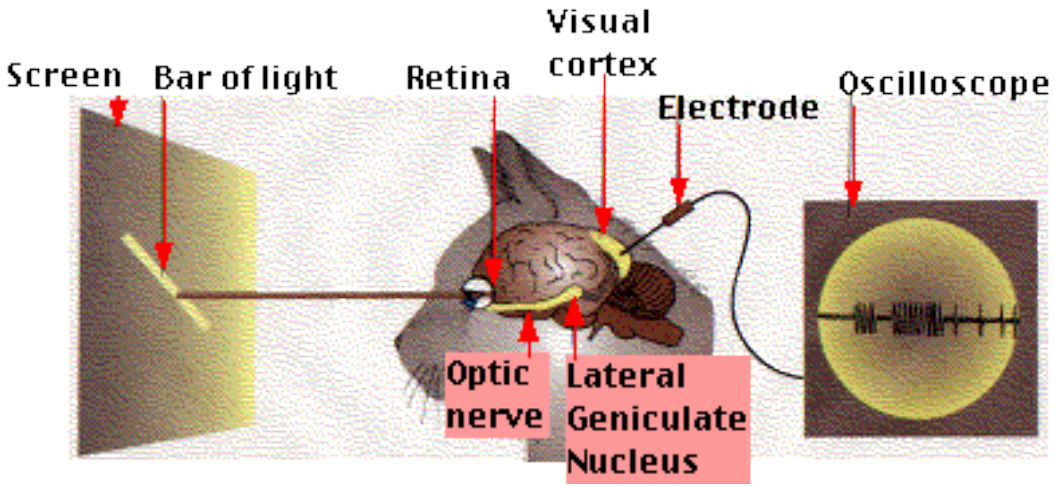


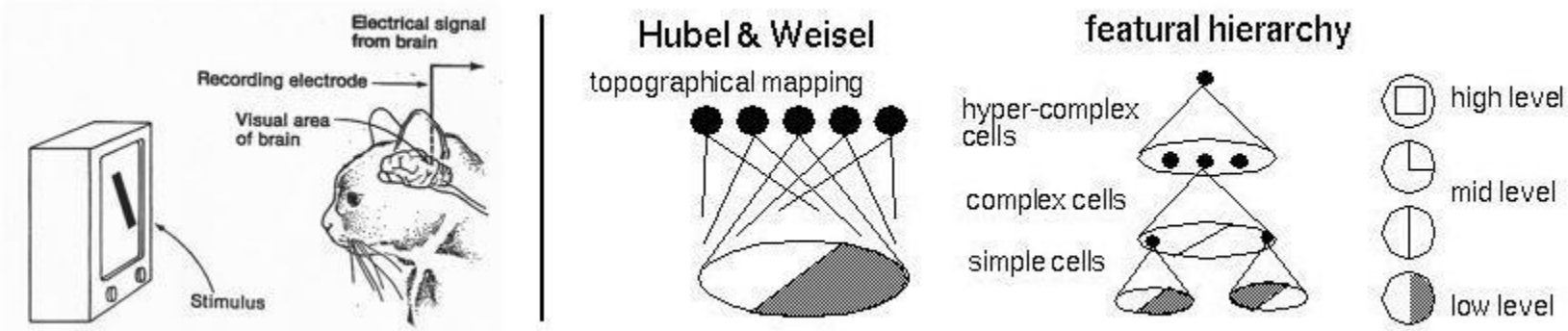




https://knowingneurons.com/2014/10/29/hubel-and-wiesel-the-neural-basis-of-visual-perception/
http://centennial.rucares.org/index.php?page=Neural_Basis_Visual_Perception

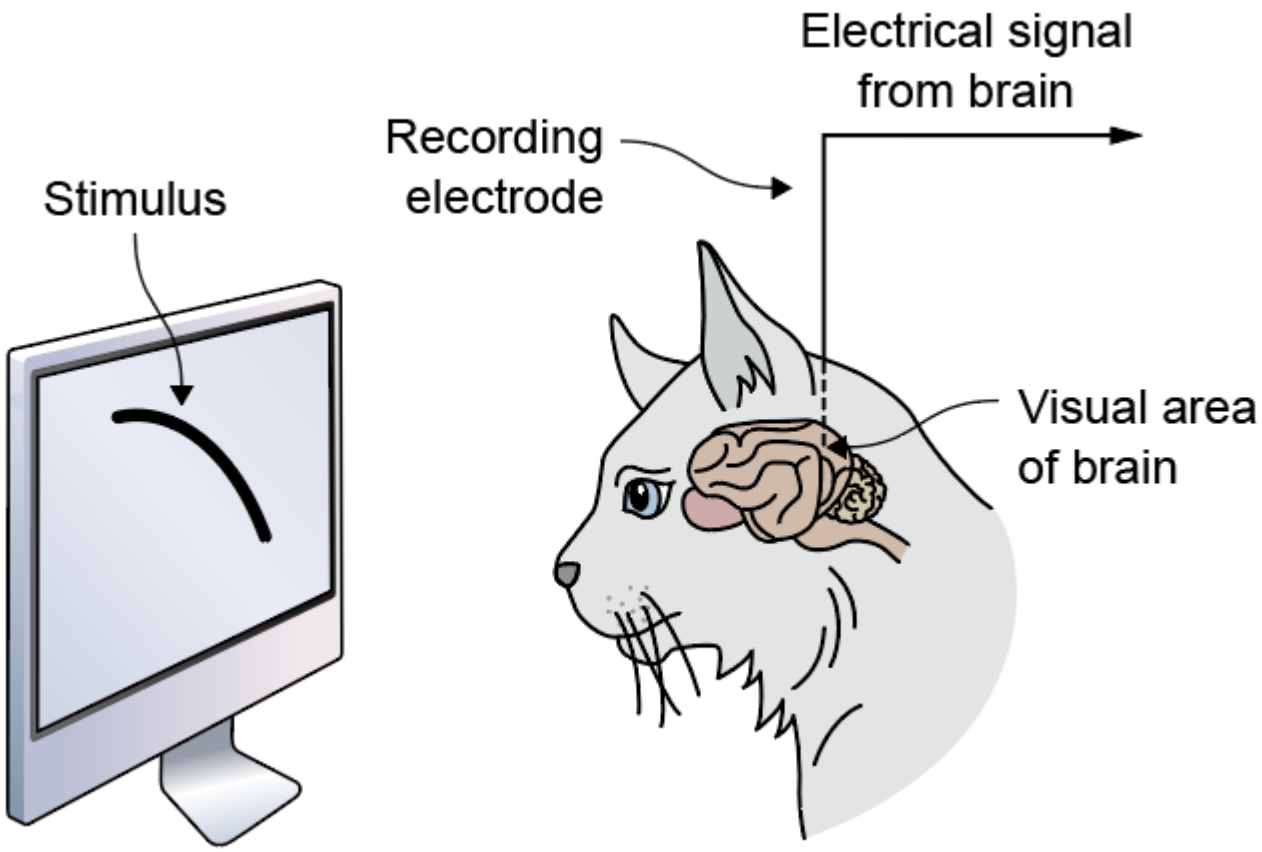

A
P30
birth
eye opening
peak CP
P0
P10
P30
B
MD
one eye covered
P0
P10
P30
C
BD
two eyes covered
P0
P10
P30


Screen
Bar of light
Retina
Visual cortex
Electrode
Oscilloscope
Optic nerve
Lateral Geniculate Nucleus



- Hubel and Weisel measured the electrical responses from a cat's brain while stimulating it with simple patterns on a television screen.
- They found was that neurons in the early visual cortex are organized in a hierarchical fashion, where the first cells connected to the cat's retinas are responsible for detecting simple patterns like edges and bars, followed by later layers responding to more complex patterns by combining the earlier neuronal activities.
- Their experiments would provide an early inspiration to artificial intelligence researchers seeking to construct well-defined computational frameworks for computer vision.

Ref : https://ml4a.github.io/ml4a/convnets/



J. Physiol. (1962), **160**, *pp.* 106–154
With 2 plates and 20 text-figures
Printed in Great Britain

RECEPTIVE FIELDS, BINOCULAR INTERACTION AND FUNCTIONAL ARCHITECTURE IN THE CAT'S VISUAL CORTEX

BY D. H. HUBEL AND T. N. WIESEL

From the Neurophysiology Laboratory, Department of Pharmacology Harvard Medical School, Boston, Massachusetts, U.S.A.

(*Received* 31 *July* 1961)

What chiefly distinguishes cerebral cortex from other parts of the central nervous system is the great diversity of its cell types and interconnexions. It would be astonishing if such a structure did not profoundly modify the response patterns of fibres coming into it. In the cat's visual cortex, the receptive field arrangements of single cells suggest that there is indeed a degree of complexity far exceeding anything yet seen at lower levels in the visual system.

In a previous paper we described receptive fields of single cortical cells, observing responses to spots of light shone on one or both retinas (Hubel & Wiesel, 1959). In the present work this method is used to examine receptive fields of a more complex type (Part I) and to make additional observations on binocular interaction (Part II).

This approach is necessary in order to understand the behaviour of individual cells, but it fails to deal with the problem of the relationship of one cell to its neighbours. In the past, the technique of recording evoked slow waves has been used with great success in studies of functional anatomy. It was employed by Talbot & Marshall (1941) and by Thompson, Woolsey & Talbot (1950) for mapping out the visual cortex in the rabbit, cat, and monkey. Daniel & Whitteridge (1959) have recently extended this work in the primate. Most of our present knowledge of retinotopic projections, binocular overlap, and the second visual area is based on these investigations. Yet the method of evoked potentials is valuable mainly for detecting behaviour common to large populations of neighbouring cells; it cannot differentiate functionally between areas of cortex smaller than about 1 mm^2. To overcome this difficulty a method has in recent years been developed for studying cells separately or in small groups during long micro-electrode penetrations through nervous tissue. Responses are correlated with cell location by reconstructing the electrode

Kunihiko Fukushima

Hubel and Wiesel's experiments inspired Kunihiko Fukushima in devising the Neocognitron, a neural network which attempted to mimic these hierarchical and compositional properties of the visual cortex.

The neocognitron was the first neural network architecture to use hierarchical layers where each layer is responsible for detecting a pattern from the output of the previous one, using a sliding filter to locate it anywhere in the image.

Biol. Cybernetics 36, 193–202 (1980)

Biological Cybernetics
© by Springer-Verlag 1980

Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position

Kunihiko Fukushima

NHK Broadcasting Science Research Laboratories, Kinuta, Setagaya, Tokyo, Japan

Abstract. A neural network model for a mechanism of visual pattern recognition is proposed in this paper. The network is self-organized by "learning without a teacher", and acquires an ability to recognize stimulus patterns based on the geometrical similarity (Gestalt) of their shapes without affected by their positions. This network is given a nickname "neocognitron". After completion of self-organization, the network has a structure similar to the hierarchy model of the visual nervous system proposed by Hubel and Wiesel. The network consists of an input layer (photoreceptor array) followed by a cascade connection of a number of modular structures, each of which is composed of two layers of cells connected in a cascade. The first layer of each module consists of "S-cells", which show characteristics similar to simple cells or lower order hypercomplex cells, and the second layer consists of "C-cells" similar to complex cells or higher order hypercomplex cells. The afferent synapses to each S-cell have plasticity and are modifiable. The network has an ability of unsupervised learning: We do not need any "teacher" during the process of self-organization, and it is only needed to present a set of stimulus patterns repeatedly to the input layer of the network. The network has been simulated on a digital computer. After repetitive presentation of a set of stimulus patterns, each stimulus pattern has become to elicit an output only from one of the C-cells of the last layer, and conversely, this C-cell has become selectively responsive only to that stimulus pattern. That is, none of the C-cells of the last layer responds to more than one stimulus pattern. The response of the C-cells of the last layer is not affected by the pattern's position at all. Neither is it affected by a small change in shape nor in size of the stimulus pattern.

1. Introduction

The mechanism of pattern recognition in the brain is little known, and it seems to be almost impossible to reveal it only by conventional physiological experiments. So, we take a slightly different approach to this problem. If we could make a neural network model which has the same capability for pattern recognition as a human being, it would give us a powerful clue to the understanding of the neural mechanism in the brain. In this paper, we discuss how to synthesize a neural network model in order to endow it an ability of pattern recognition like a human being.

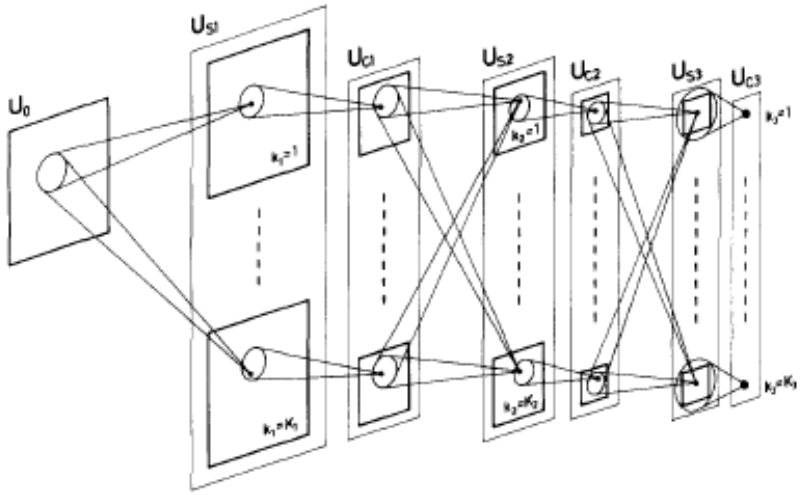
Several models were proposed with this intention (Rosenblatt, 1962; Kabrisky, 1966; Giebel, 1971; Fukushima, 1975). The response of most of these models, however, was severely affected by the shift in position and/or by the distortion in shape of the input patterns. Hence, their ability for pattern recognition was not so high.

In this paper, we propose an improved neural network model. The structure of this network has been suggested by that of the visual nervous system of the vertebrate. This network is self-organized by "learning without a teacher", and acquires an ability to recognize stimulus patterns based on the geometrical similarity (Gestalt) of their shapes without affected by their position nor by small distortion of their shapes.

This network is given a nickname "neocognitron"[1], because it is a further extention of the "cognitron", which also is a self-organizing multilayered neural network model proposed by the author before (Fukushima, 1975). Incidentally, the conventional cognitron also had an ability to recognize patterns, but its response was dependent upon the position of the stimulus patterns. That is, the same patterns which were presented at different positions were taken as different patterns by the conventional cognitron. In the neocognitron proposed here, however, the response of the network is little affected by the position of the stimulus patterns.

1 Preliminary report of the neocognitron already appeared elsewhere (Fukushima, 1979a, b)

0340-1200/80/0036/0193/$02.00



Schematic diagram illustrating the interconnections between layers in the neocognitron

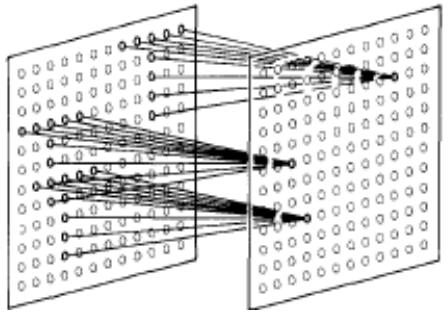


Illustration showing the input interconnections to the cells within a single cell-plane

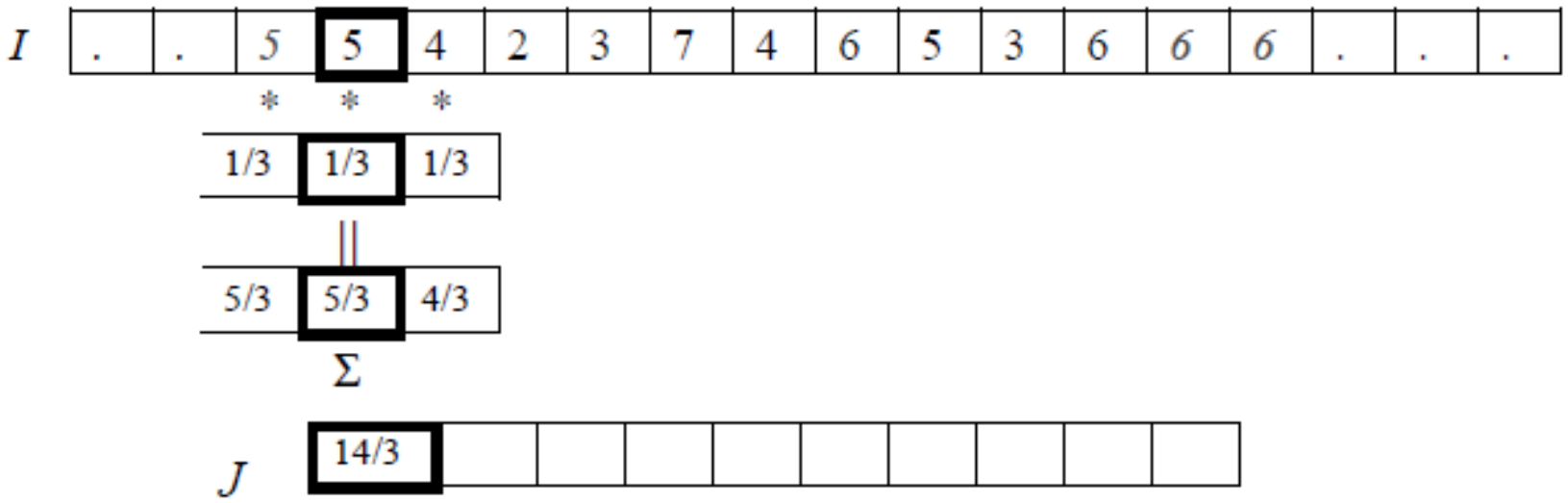
- Although the neocognitron achieved some success in pattern recognition tasks and introduced convolutional filters to neural networks, it was limited by its lack of a training algorithm to learn the filters.
- The pattern detectors were manually engineered for the specific task, using a variety of heuristics and techniques from computer vision
- Backpropagation had not yet been applied to train neural nets, and thus there was no easy way to optimize neocognitrons or reuse them on different vision tasks.

Convolution

- Convolution is a basic operation that is used to extract information from images.
- It is simple, can be analyzed and understood very well, and is also easy to implement and can be computed very efficiently.
- It has two key features: it is *shift-invariant, and linear.*
- Shift-invariant means that we perform the same operation at every point in the image.
- Linear means that this operation is linear, that is, we replace every pixel with a linear combination of its neighbors

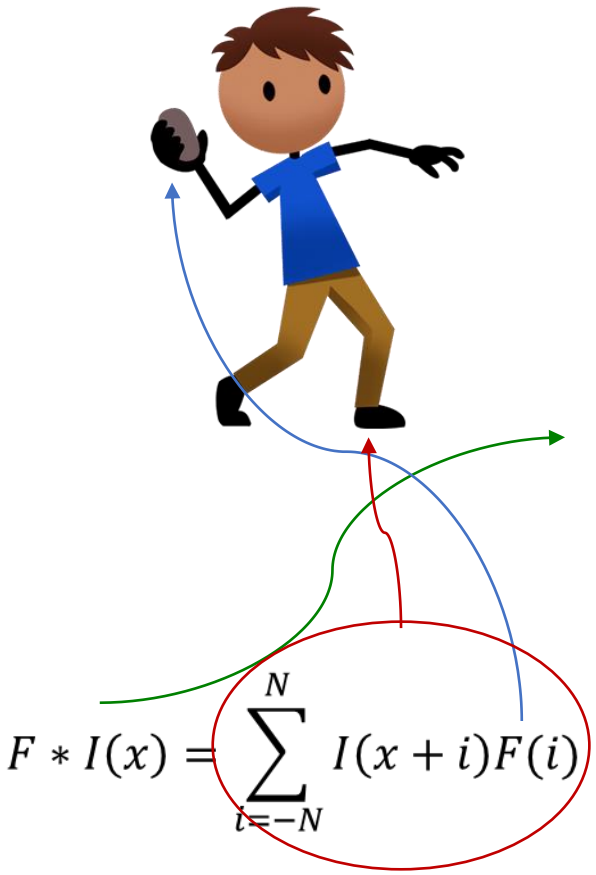
One of the simplest operations that we can perform with convolution is local averaging (The convolution that we are discussing here is correlation in the image processing literature).

5	4	2	3	7	4	6	5	3	6



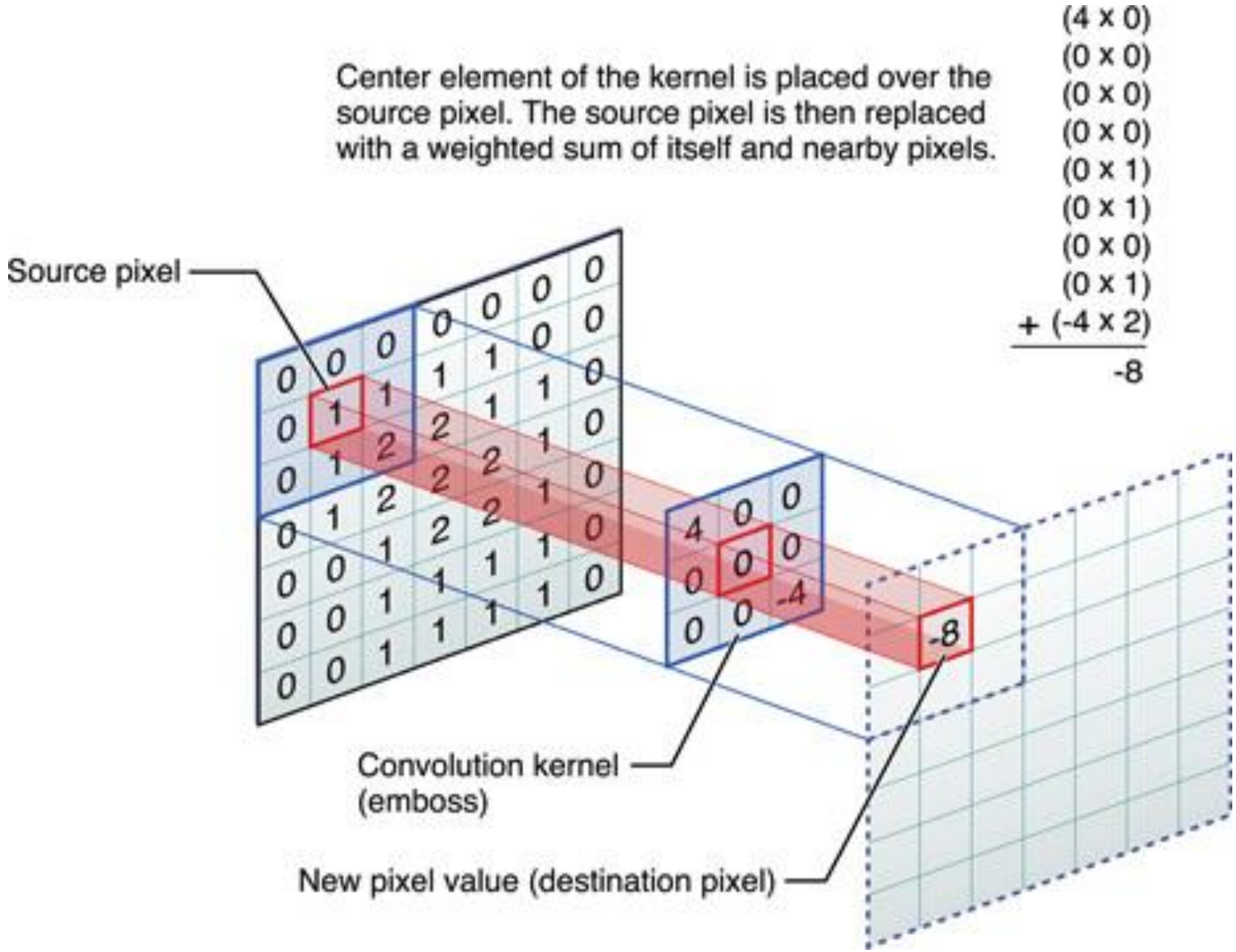
$$F * I(x) = \sum_{i=-N}^{N} I(x+i)F(i)$$

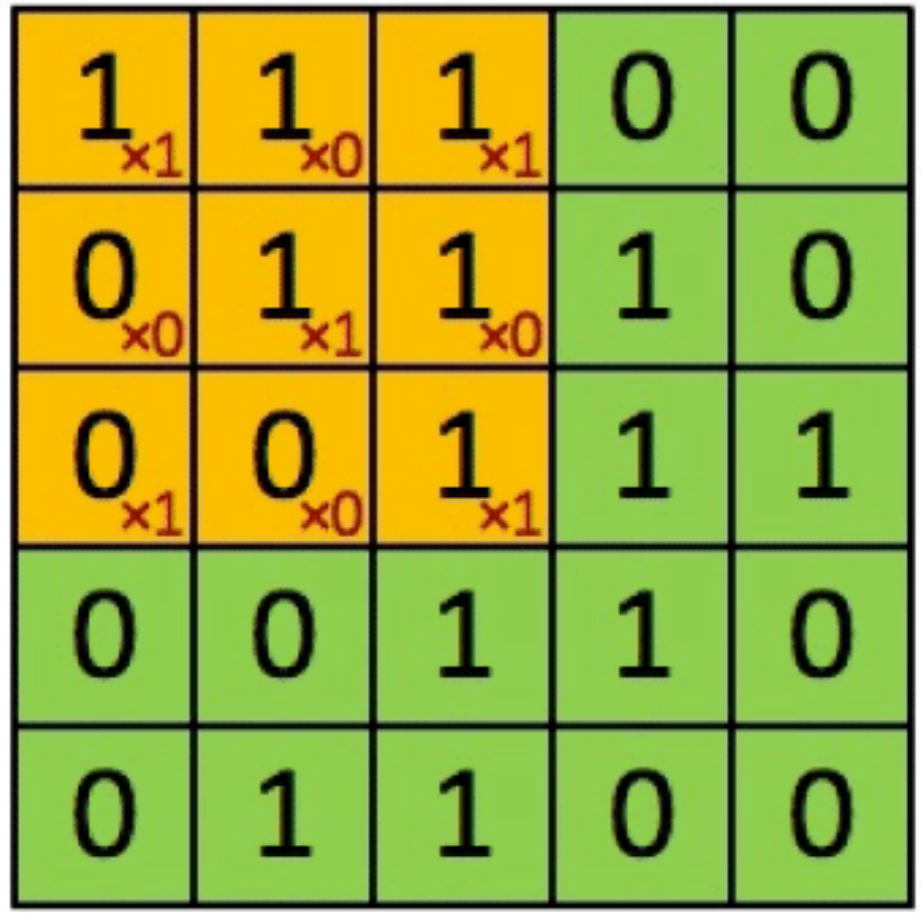
$$F * I(x,y) = \sum_{j=N}^{N} \sum_{i=-N}^{N} I(x+i,y+j)F(i,j)$$



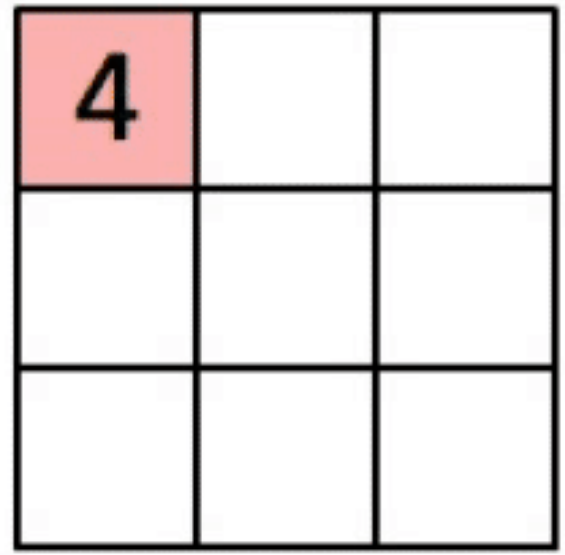
$$F * I(x) = \sum_{i=-N}^{N} I(x+i)F(i)$$

Center element of the kernel is placed over the source pixel. The source pixel is then replaced with a weighted sum of itself and nearby pixels.





Image



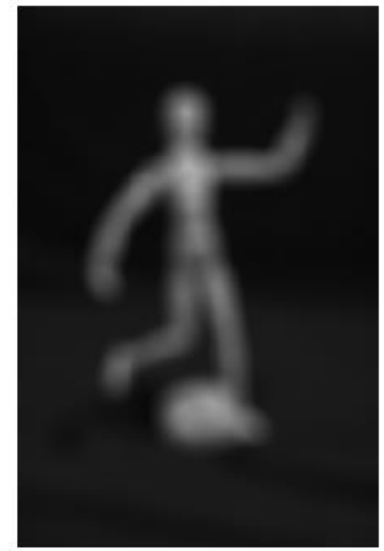
Convolved Feature

Mean Filters: Effect of Filter Size

Original

7×7

15×15



41×41

Kernel Size: The kernel size defines the field of view of the convolution. A common choice for 2D is 3 — that is 3x3 pixels.

Stride: The stride defines the step size of the kernel when traversing the image.

Padding: The padding defines how the border of a sample is handled.

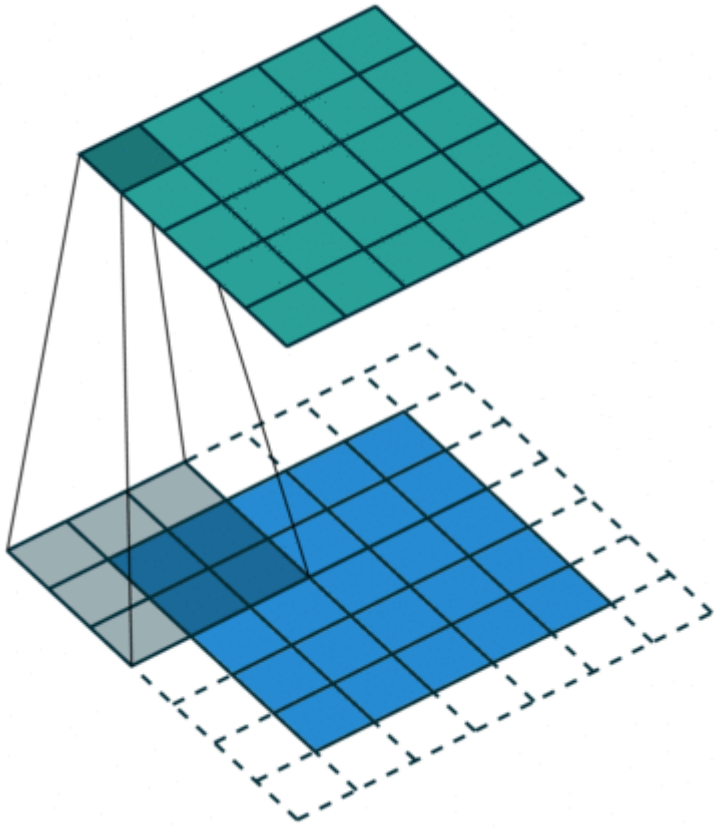


Image Size after convolution

Image size : n x n , filter size : f x f

Image size after convolution : $(n - f + 1) \times (n - f + 1)$

With zero padding: $(n + 2p - f + 1) \times (n + 2p - f + 1)$

Strided convolution: $\left(\left\lfloor \frac{n + 2p - f}{s} + 1 \right\rfloor \times \left\lfloor \frac{n + 2p - f}{s} + 1 \right\rfloor\right)$

❑ Conv 3x3 with stride=1,padding=0

1	2	1	1	1	1
1	1	5	3	9	1
2	4	4	7	5	3
3	6	7	5	6	2
1	6	5	3	1	2
2	3	1	1	1	1

6x6 Image

21	28	36	31
33	42	51	41
38	47	43	34
34	37	30	22

4x4

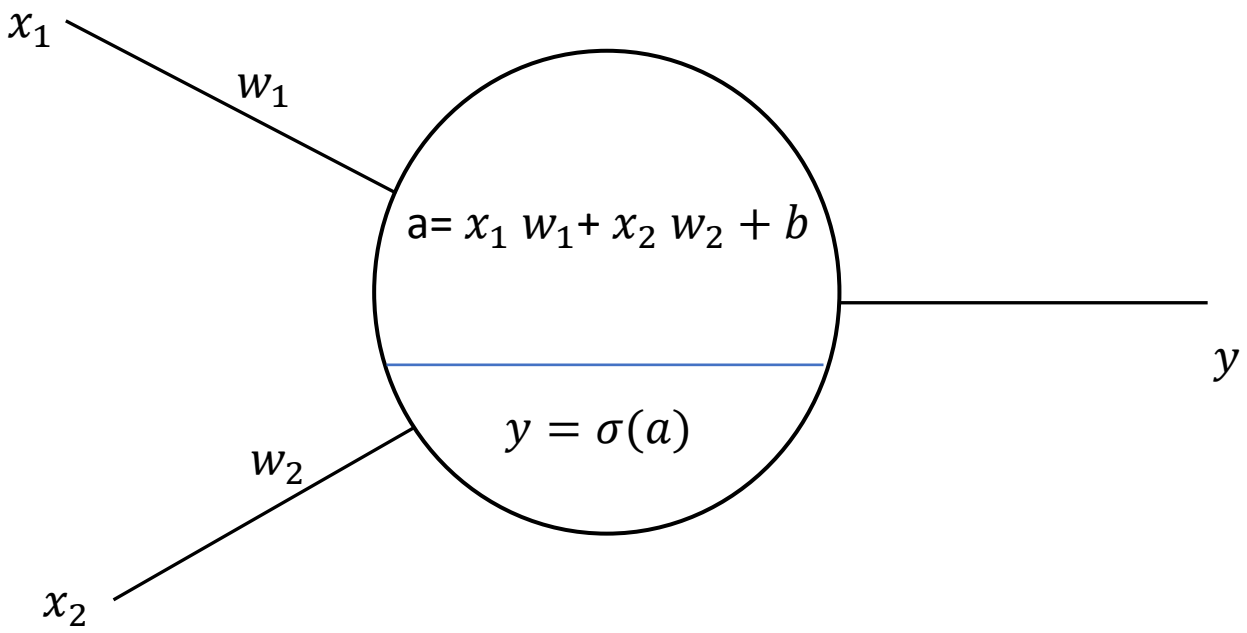
❑ Conv 3x3 with **stride=2**,padding=0

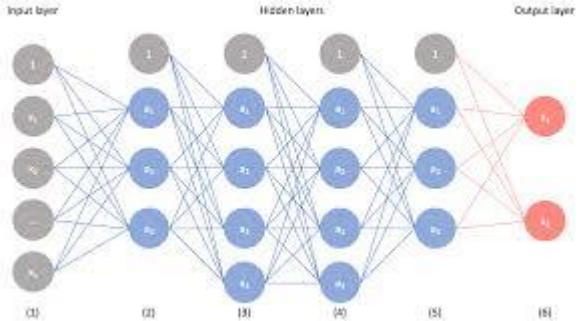
1	2	1	1	1	1	1
1	1	5	3	9	1	1
2	4	4	7	5	3	1
3	6	7	5	6	2	2
1	6	5	3	1	2	1
2	3	1	1	1	1	1
1	1	1	3	2	2	1

7x7 Image

14	36	23
38	43	23
21	18	12

3x3


x_1
w_1
a= $x_1\ w_1 + x_2\ w_2 + b$
$y = \sigma(a)$
y
w_2
x_2


Input layer
Hidden layers
Output layer

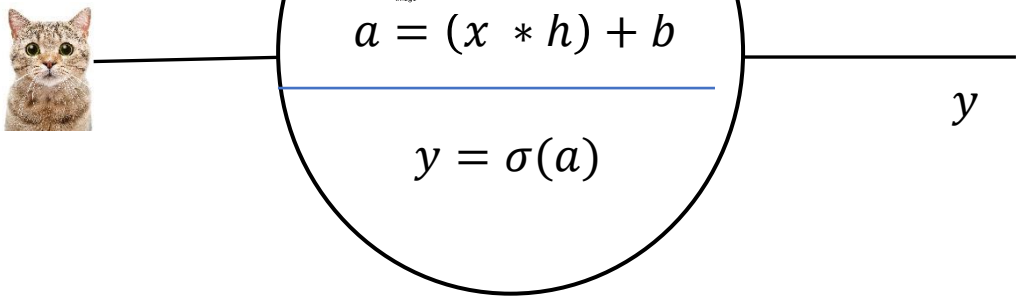
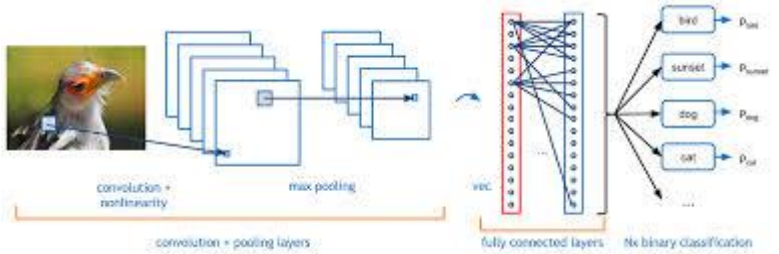
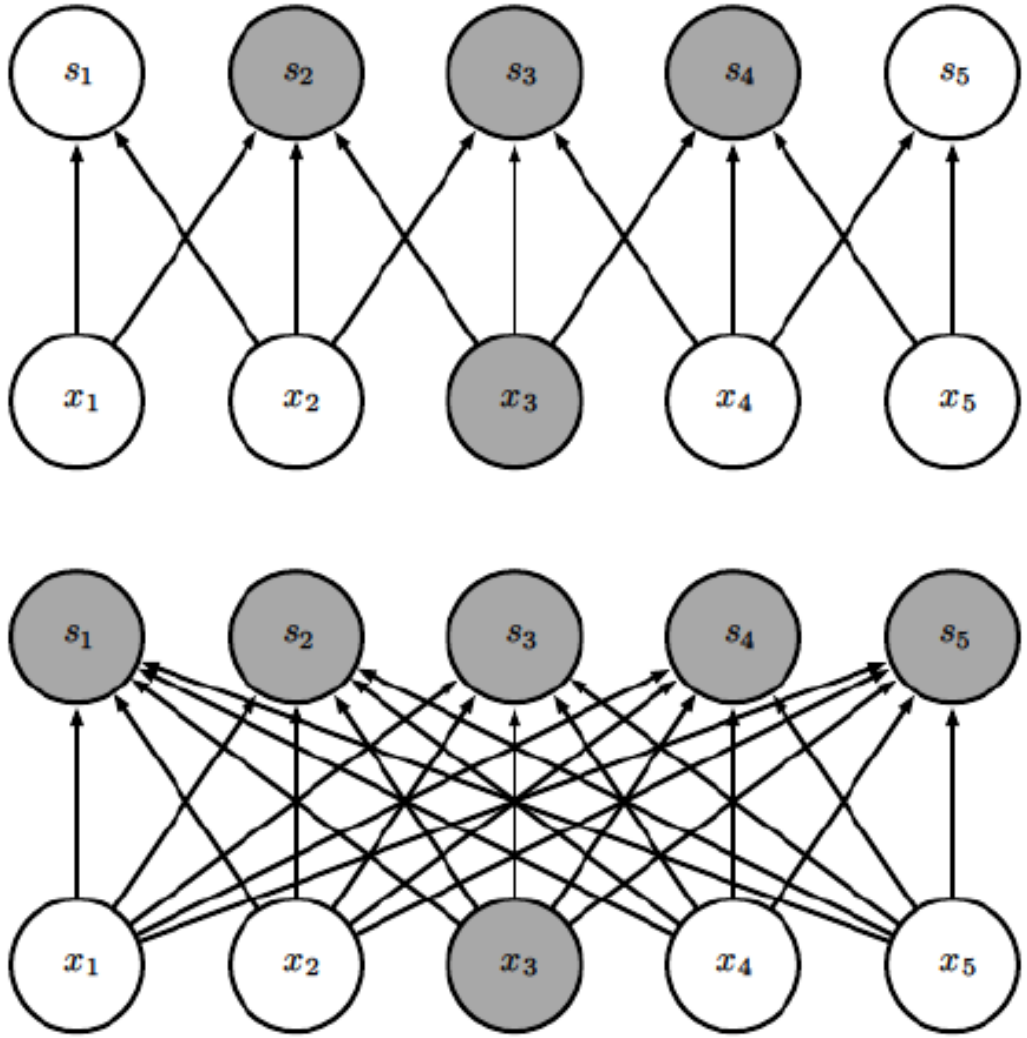


Image
Filter / Kernel
Feature
$a = (x * h) + b$
$y = \sigma(a)$
y


convolution + nonlinearity
max pooling
convolution + pooling layers
vec
fully connected layers
bird
sunset
dog
cat
fix binary classification



Number of parameters will be very less compared to conventional ANN

Figure 9.2: *Sparse connectivity, viewed from below:* We highlight one input unit, x_3, and also highlight the output units in $\boldsymbol{s}$ that are affected by this unit. *(Top)* When $\boldsymbol{s}$ is formed by convolution with a kernel of width 3, only three outputs are affected by $\boldsymbol{x}$. *(Bottom)* When $\boldsymbol{s}$ is formed by matrix multiplication, connectivity is no longer sparse, so all of the outputs are affected by x_3.

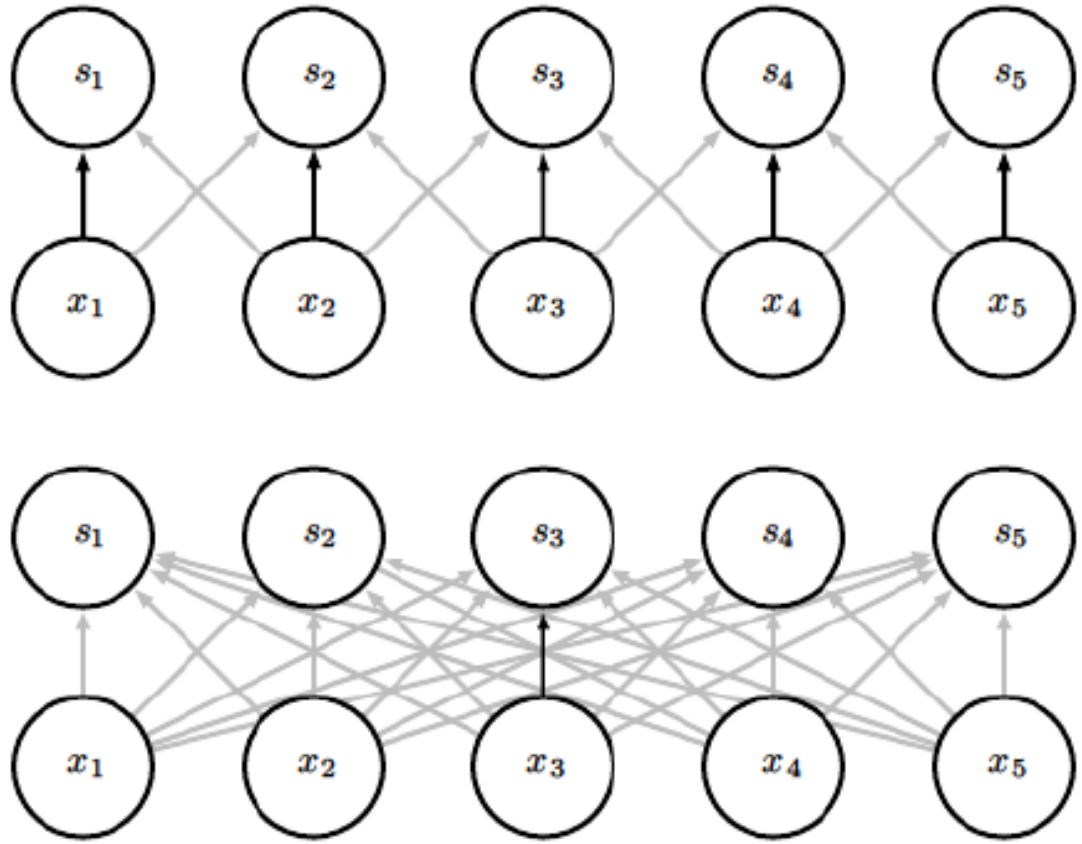
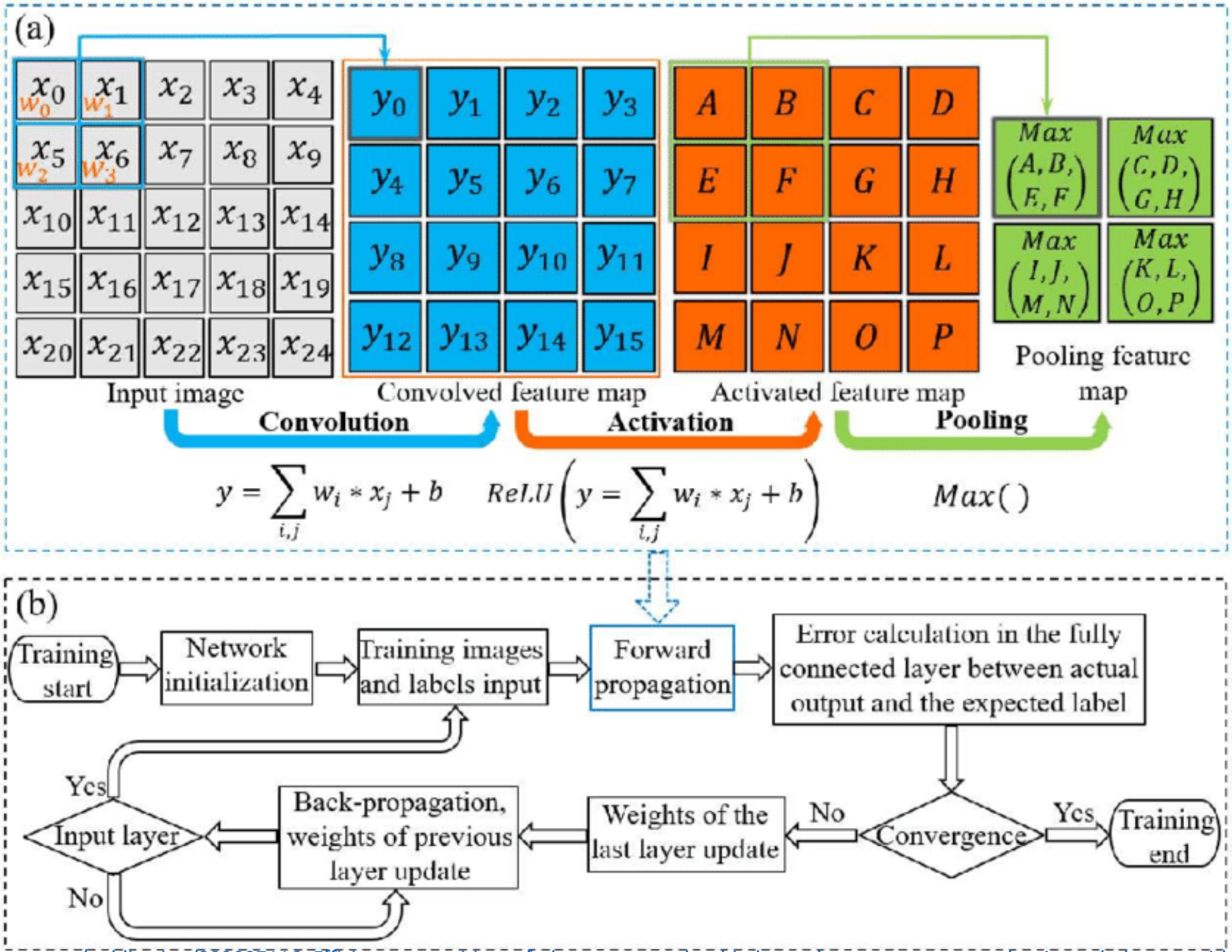
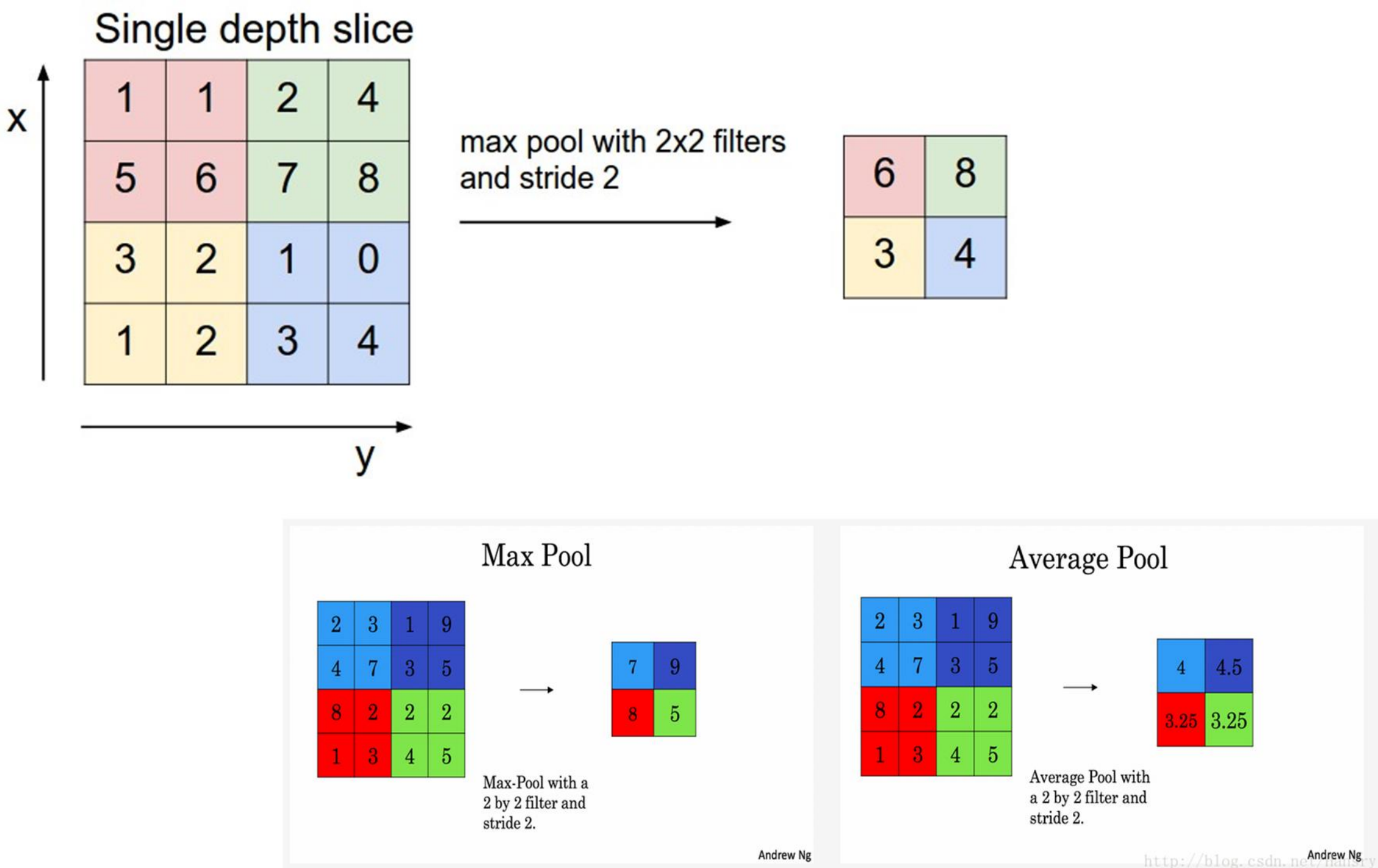


Figure 9.5: *Parameter sharing:* Black arrows indicate the connections that use a particular parameter in two different models. *(Top)* The black arrows indicate uses of the central element of a 3-element kernel in a convolutional model. Due to parameter sharing, this single parameter is used at all input locations. *(Bottom)* The single black arrow indicates the use of the central element of the weight matrix in a fully connected model. This model has no parameter sharing so the parameter is used only once.



https://www.researchgate.net/publication/333007116_Efficient_recognition_of_the_propagated_orbital_angular_momentum_modes_in_turbulences_with_the_convolutional_neural_network

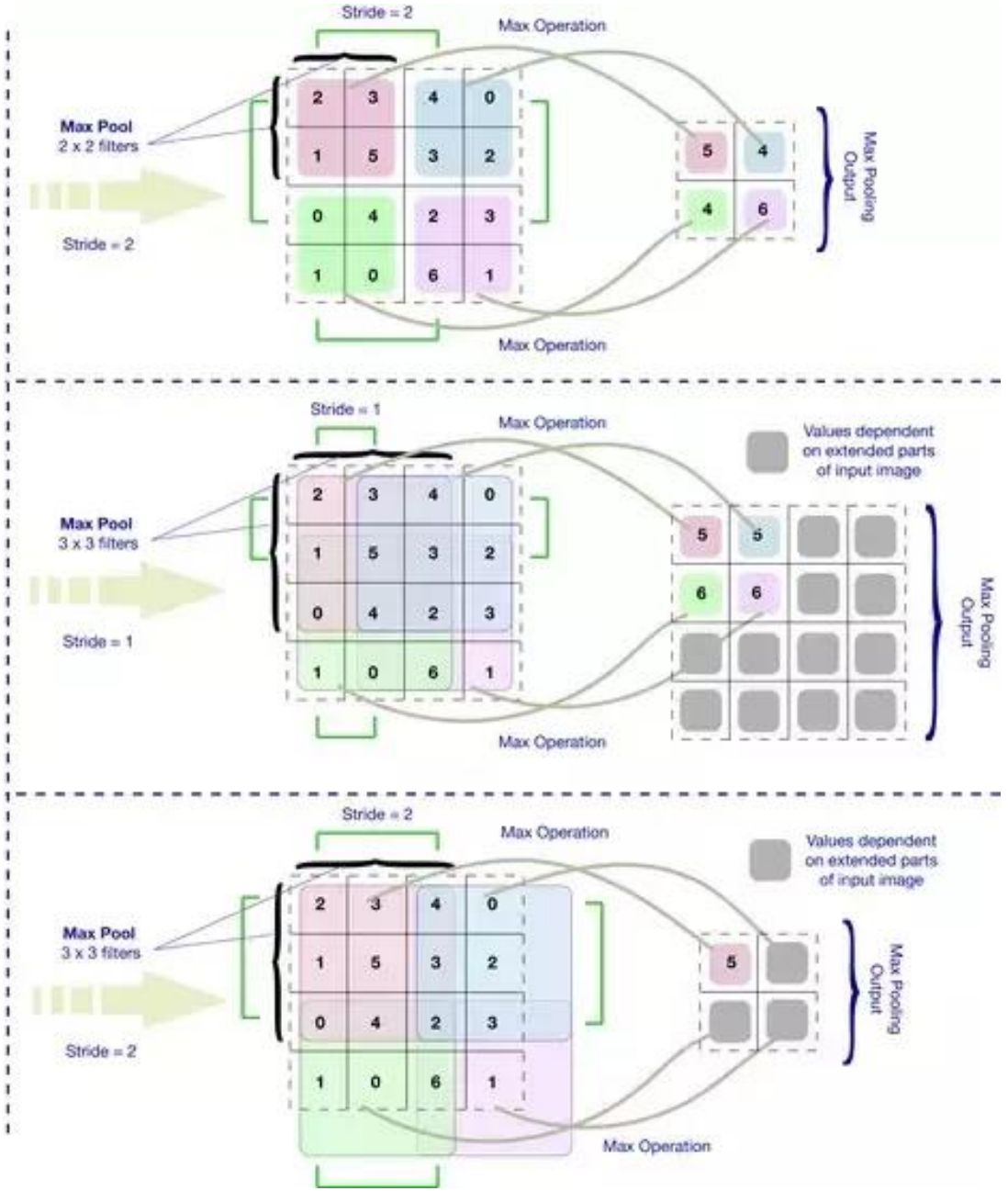


http://blog.csdn.net/hansry

Image size : n x n , filter size : f x f

Image size after pooling : $\left(\frac{n-f}{s}+1 \times \frac{n-f}{s}+1\right)$

- The purpose of pooling layers is to perform dimensionality reduction to widen subsequent convolutional layers' receptive fields.
- The same effect can be achieved by using a convolutional layer: using a stride of 2 also reduces the dimensionality of the output and widens the receptive field of higher layers.
- The resulting operation differs from a max-pooling layer in that
  - –it cannot perform a true max operation
  - –it allows pooling across input channels



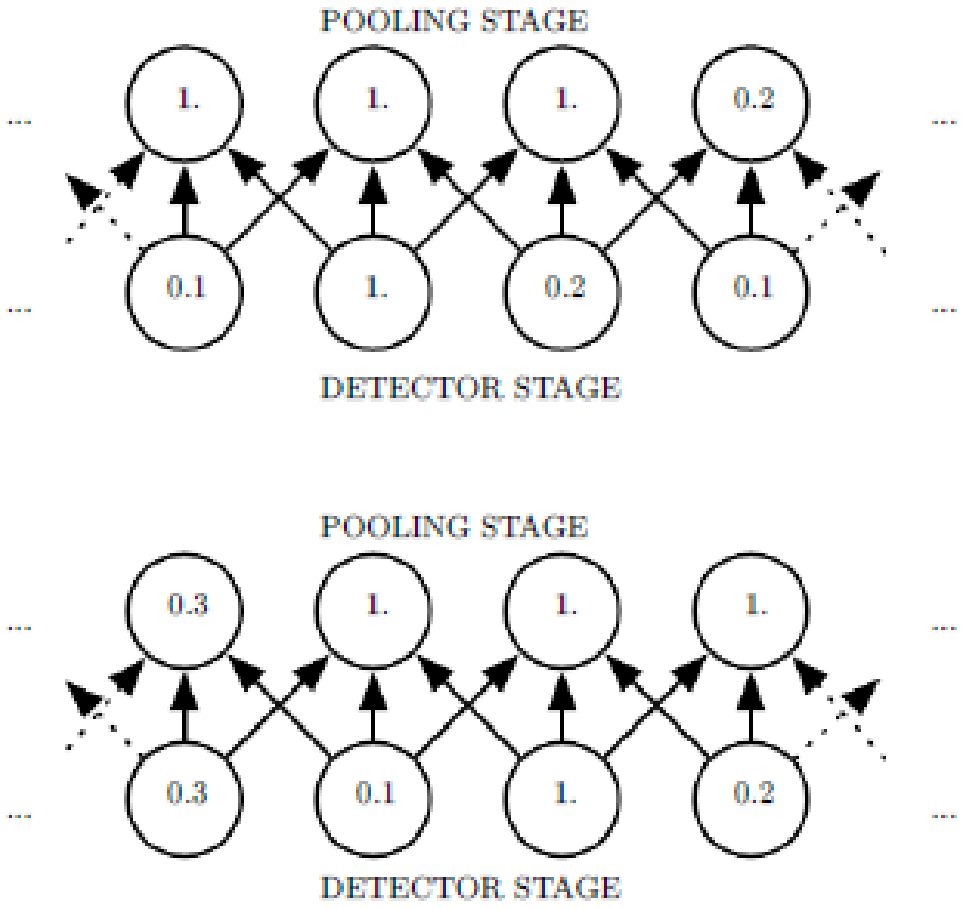


Figure 9.8: Max pooling introduces invariance. *(Top)* A view of the middle of the output of a convolutional layer. The bottom row shows outputs of the nonlinearity. The top row shows the outputs of max pooling, with a stride of one pixel between pooling regions and a pooling region width of three pixels. *(Bottom)* A view of the same network, after the input has been shifted to the right by one pixel. Every value in the bottom row has changed, but only half of the values in the top row have changed, because the max pooling units are only sensitive to the maximum value in the neighborhood, not its exact location.

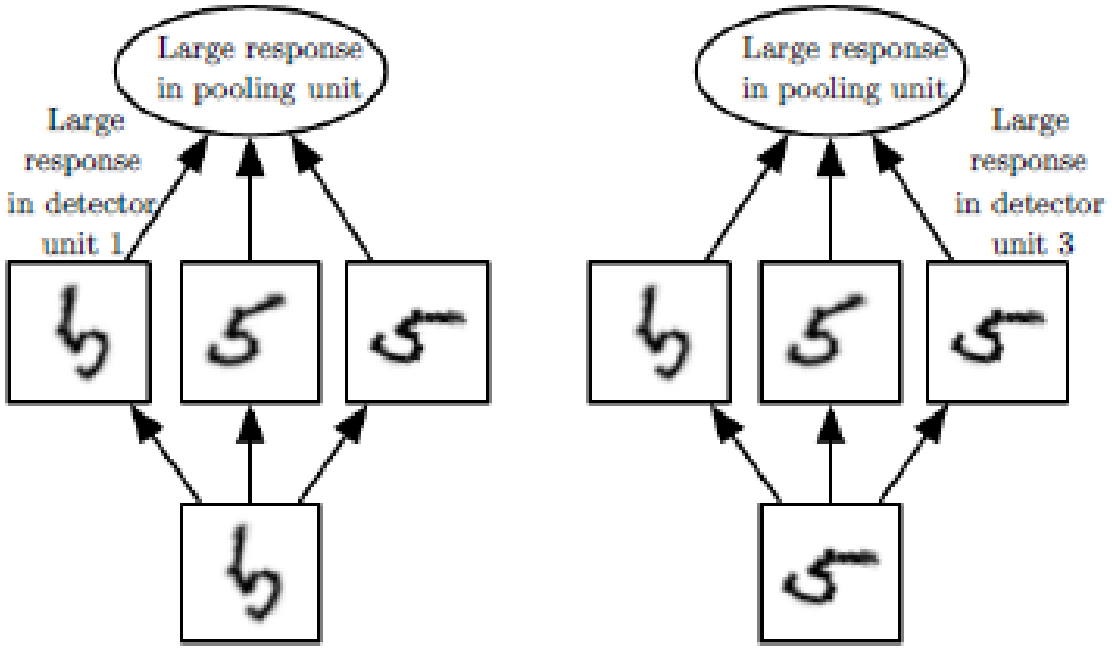
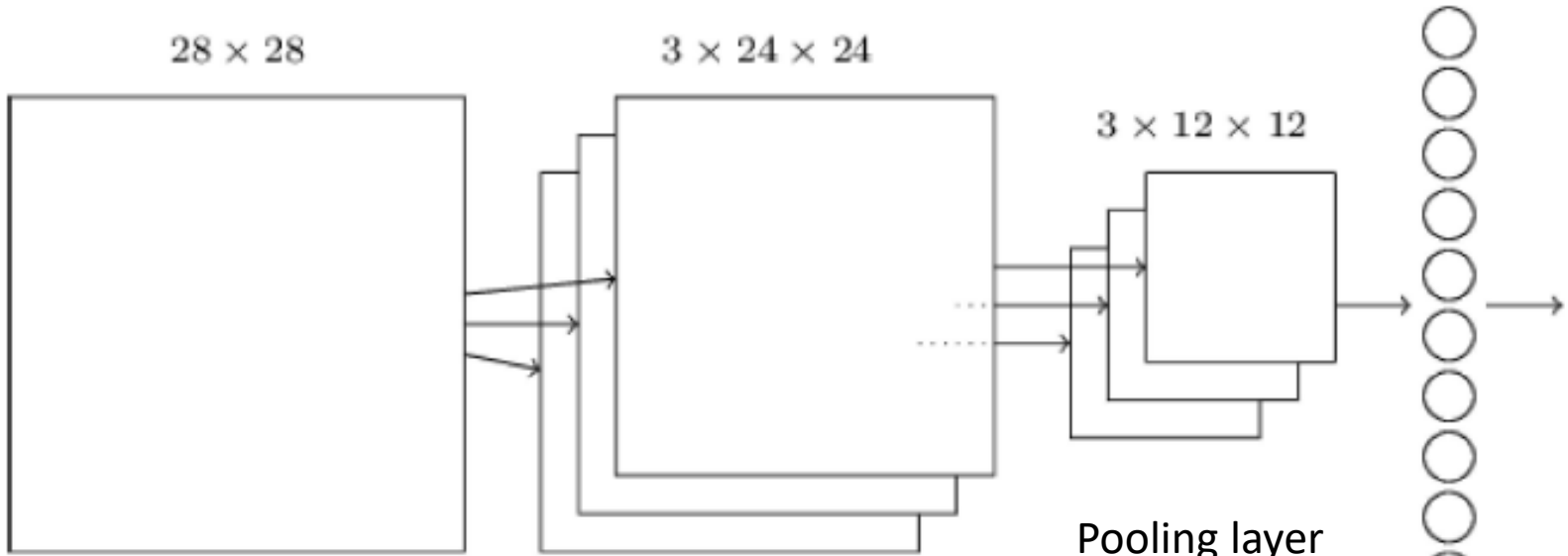
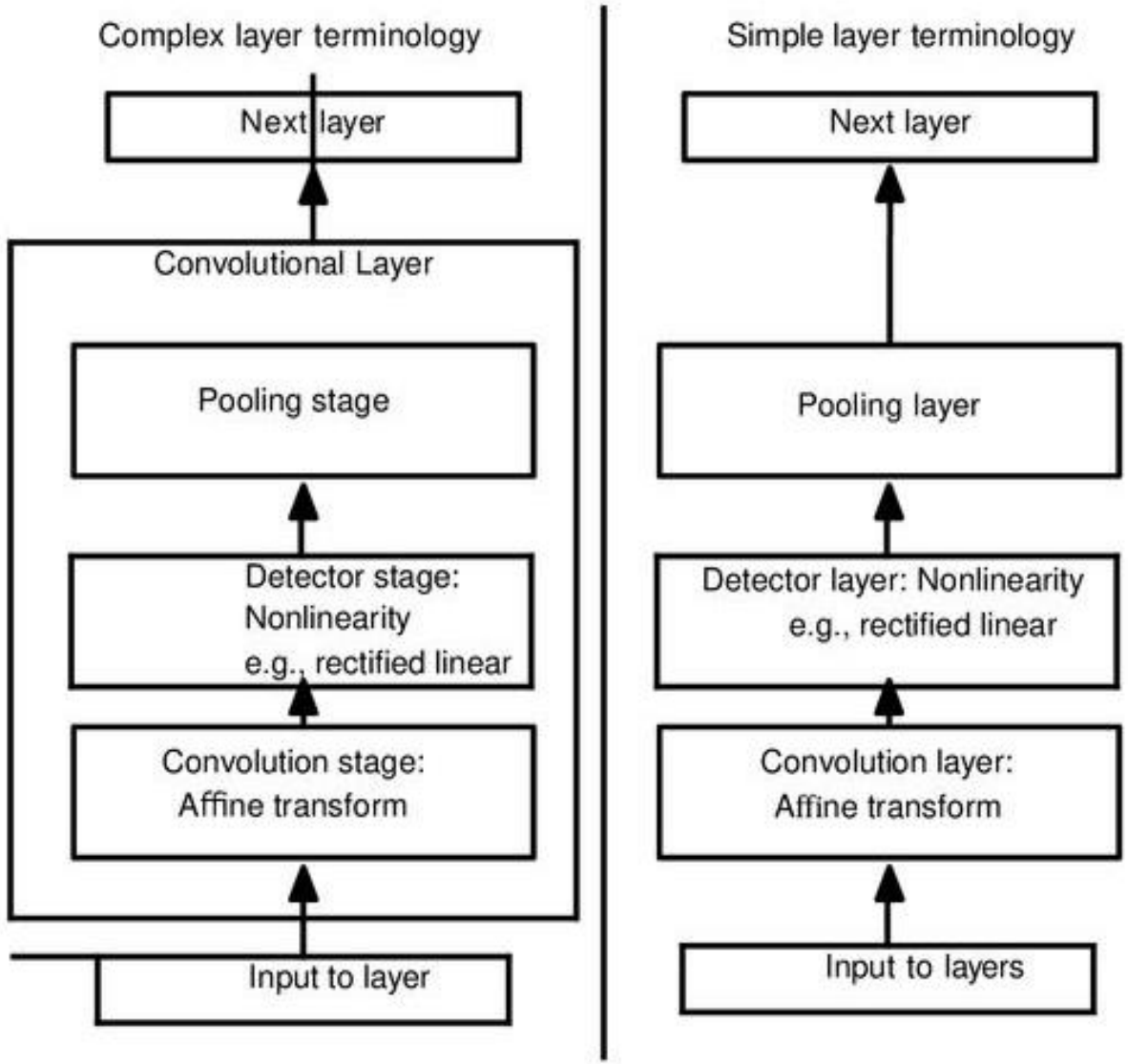


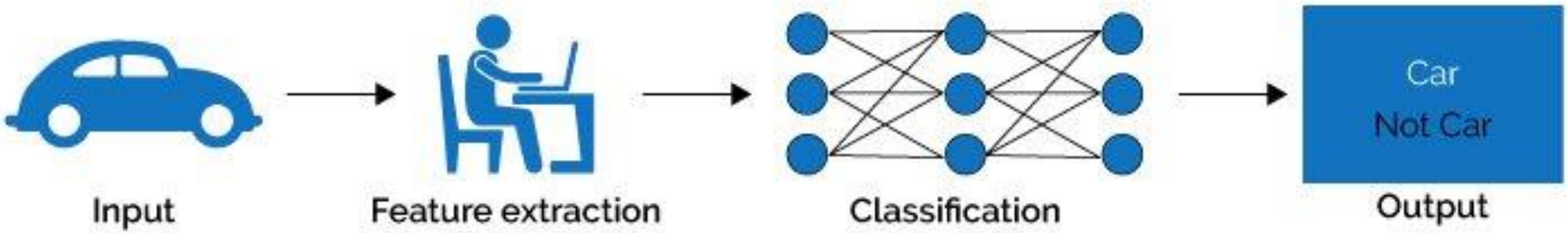
Figure 9.9: *Example of learned invariances:* A pooling unit that pools over multiple features that are learned with separate parameters can learn to be invariant to transformations of the input. Here we show how a set of three learned filters and a max pooling unit can learn to become invariant to rotation. All three filters are intended to detect a hand-written 5. Each filter attempts to match a slightly different orientation of the 5. When a 5 appears in the input, the corresponding filter will match it and cause a large activation in a detector unit. The max pooling unit then has a large activation regardless of which pooling unit was activated. We show here how the network processes two different inputs, resulting in two different detector units being activated. The effect on the pooling unit is roughly the same either way. This principle is leveraged by maxout networks (Goodfellow *et al.*, 2013a) and other convolutional networks. Max pooling over spatial positions is naturally invariant to translation; this multi-channel approach is only necessary for learning other transformations.


28 × 28
3 × 24 × 24
3 × 12 × 12
Input layer
Convolutional layer
Pooling layer
Fully connected layer

Two common terminologies

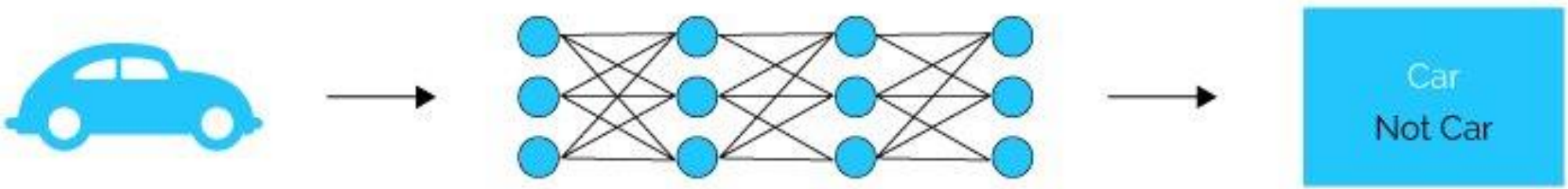


Machine Learning



Input → Feature extraction → Classification → Output

Deep Learning



Input → Feature extraction + Classification → Output