Convolution in Machine Learning

劉晉良

Jinn-Liang Liu

清華大學計算與建模科學研究所

Institute of Computational and Modeling Science

National Tsing Hua University, Taiwan

Oct 26, 2018

Source: Convolution by Song Ho Ahn

Definition (in Discrete Time)

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n-k]$$

time-shifted

$x[n]$ is input signal, $h[n]$ is impulse response,

$y[n]$ is output. * denotes convolution.

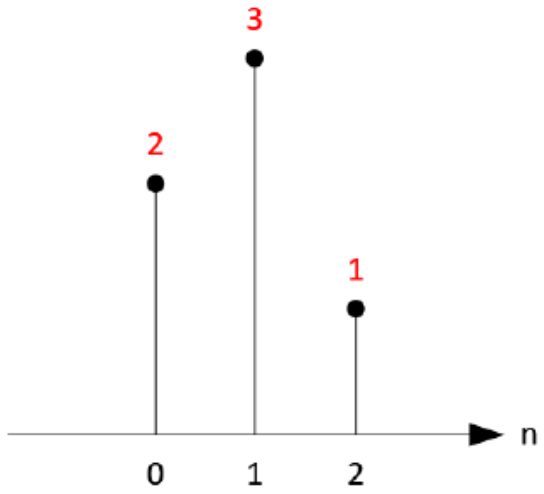
impulse response impulse decomposition

Impulse Function Decomposition

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots \qquad f(x) = \sum_{n=0}^{\infty} a_n (x-b)^n$$

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \cdots \qquad x[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot \delta[n-k]$$

In general, a signal can be decomposed as a weighted sum of basis signals.



$$x[0] = x[0] \cdot \delta[n] \quad = 2 \cdot \delta[n-0]$$
$$x[1] = x[1] \cdot \delta[n-1] = 3 \cdot \delta[n-1]$$
$$x[2] = x[2] \cdot \delta[n-2] = 1 \cdot \delta[n-2]$$

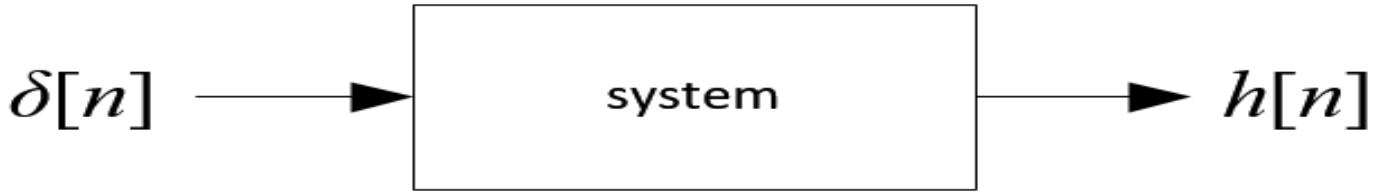
a signal is decomposed into a set of impulse (delta) functions

$\delta[n]$ is 1 at $n=0$

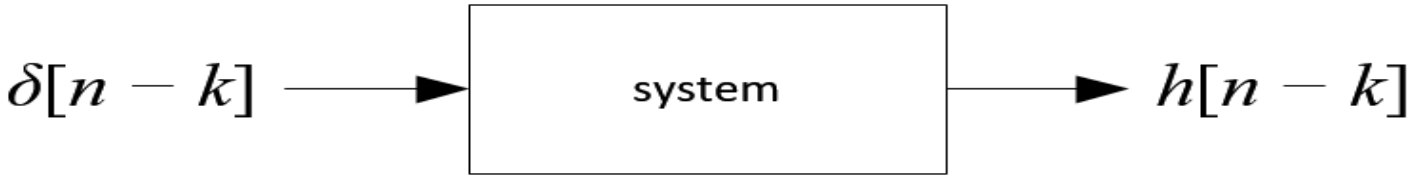
zeros at $n \neq 0$

$$x[n] = x[0] \cdot \delta[n-0] + x[1] \cdot \delta[n-1] + x[2] \cdot \delta[n-2]$$

Impulse Response

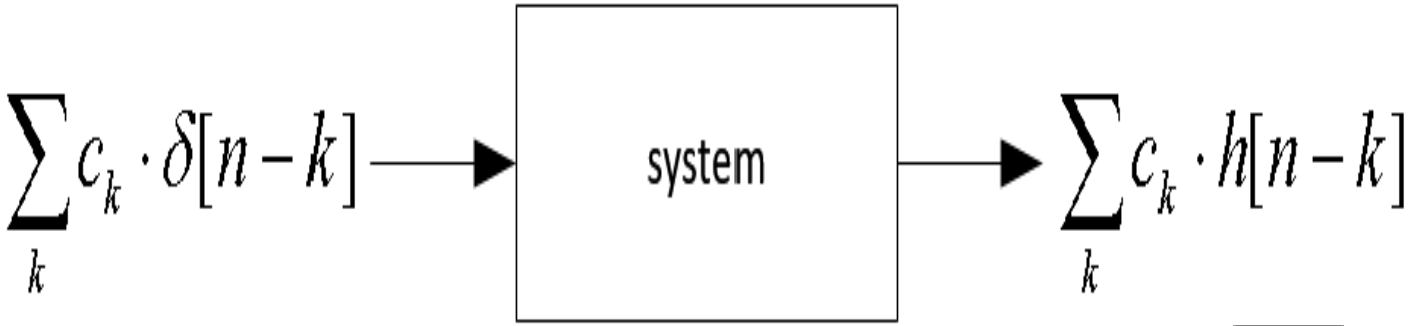


impulse response $h[n]$



the system is time-invariant

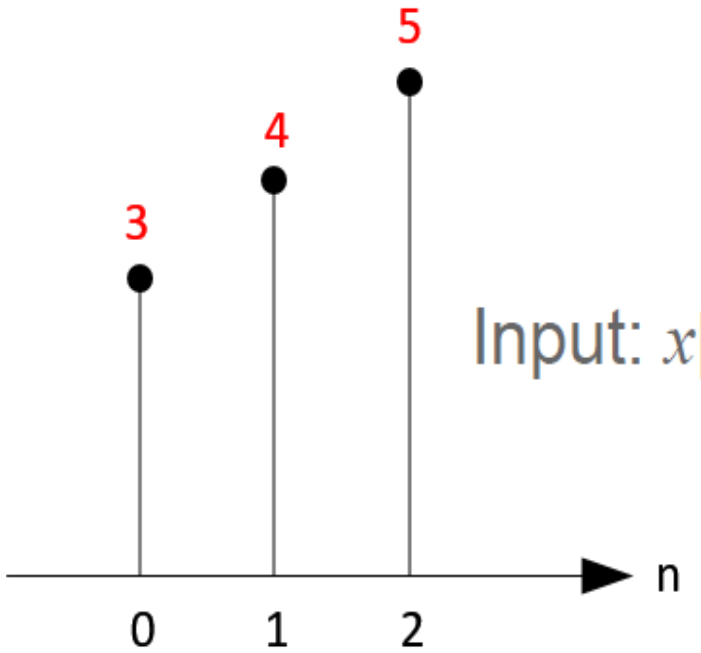
the system is linear

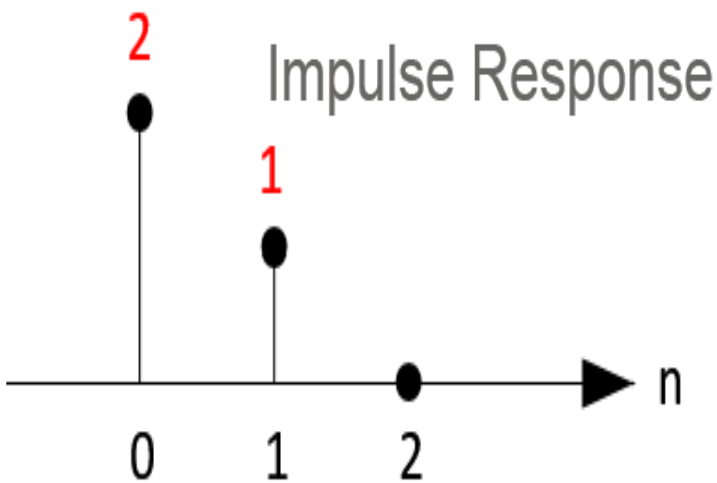


$$x[n] = \sum x[k] \cdot \delta[n-k]$$

$$y[n] = \sum x[k] \cdot h[n-k]$$

Convolution in 1D

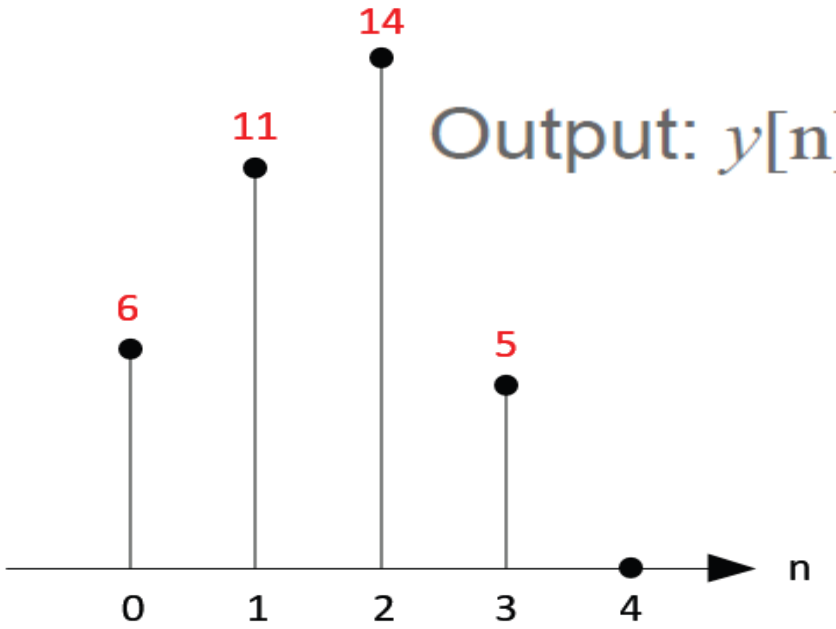




$$
\begin{aligned}
y[0] &= \sum_{k=-\infty}^{\infty} x[k] \cdot h[0-k] \\
&= x[0] \cdot h[0] + x[1] \cdot h[0-1] + x[2] \cdot h[0-2] + \dots \\
&= x[0] \cdot h[0] \\
&= 3 \cdot 2 \\
&= 6
\end{aligned}
$$

$$
\begin{aligned}
y[1] &= \sum_{k=-\infty}^{\infty} x[k] \cdot h[1-k] \\
&= x[0] \cdot h[1-0] + x[1] \cdot h[1-1] + x[2] \cdot h[1-2] + \dots \\
&= x[0] \cdot h[1] + x[1] \cdot h[0] \\
&= 3 \cdot 1 + 4 \cdot 2 \\
&= 11
\end{aligned}
$$

$$
\begin{aligned}
y[2] &= \sum_{k=-\infty}^{\infty} x[k] \cdot h[2-k] \\
&= x[0] \cdot h[2-0] + x[1] \cdot h[2-1] + x[2] \cdot h[2-2] + \dots \\
&= x[0] \cdot h[2] + x[1] \cdot h[1] + x[2] \cdot h[0] \\
&= 3 \cdot 0 + 4 \cdot 1 + 5 \cdot 2 \\
&= 14
\end{aligned}
$$



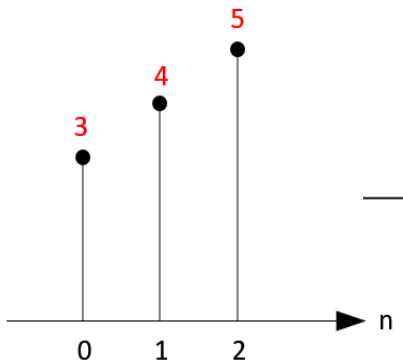
$$y[3] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[3-k]$$

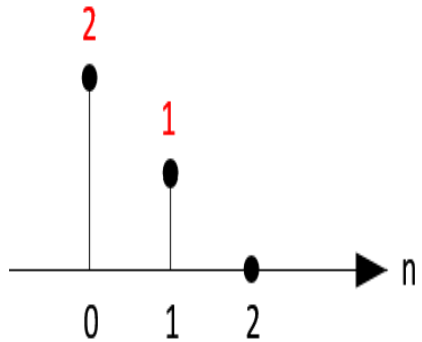
$$= x[0] \cdot h[3-0] + x[1] \cdot h[3-1] + x[2] \cdot h[3-2] + x[3] \cdot h[3-3] + \ldots$$

$$= x[0] \cdot h[3] + x[1] \cdot h[2] + x[2] \cdot h[1] + x[3] \cdot h[0]$$

$$= 3 \cdot 0 + 4 \cdot 0 + 5 \cdot 1 + 0 \cdot 2$$

$$= 5$$





$$y[0] = x[0] \cdot h[0]$$
$$y[1] = x[1] \cdot h[0] + x[0] \cdot h[1]$$
$$y[2] = x[2] \cdot h[0] + x[1] \cdot h[1]$$
$$y[3] = x[3] \cdot h[0] + x[2] \cdot h[1]$$

$$y[4] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[4-k]$$

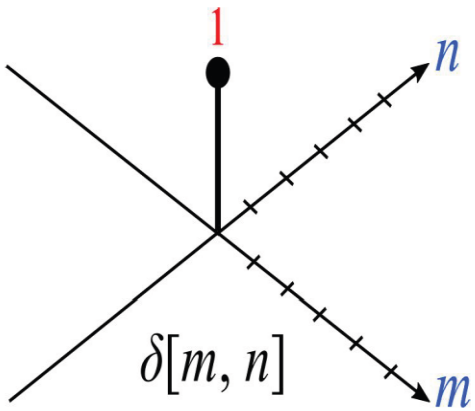
$$= x[0] \cdot h[4-0] + x[1] \cdot h[4-1] + x[2] \cdot h[4-2] + x[3] \cdot h[4-3] + x[4] \cdot h[4-4] + ..$$

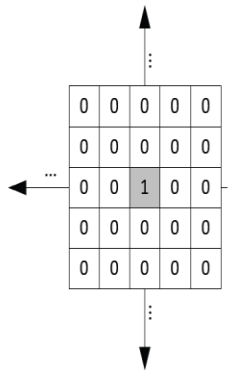
$$= x[0] \cdot h[4] + x[1] \cdot h[3] + x[2] \cdot h[2] + x[3] \cdot h[1] + x[4] \cdot h[0]$$

$$= 3 \cdot 0 + 4 \cdot 0 + 5 \cdot 0 + 0 \cdot 2 + 0 \cdot 1 + 0 \cdot 0$$

$$= 0$$

Convolution in 2D



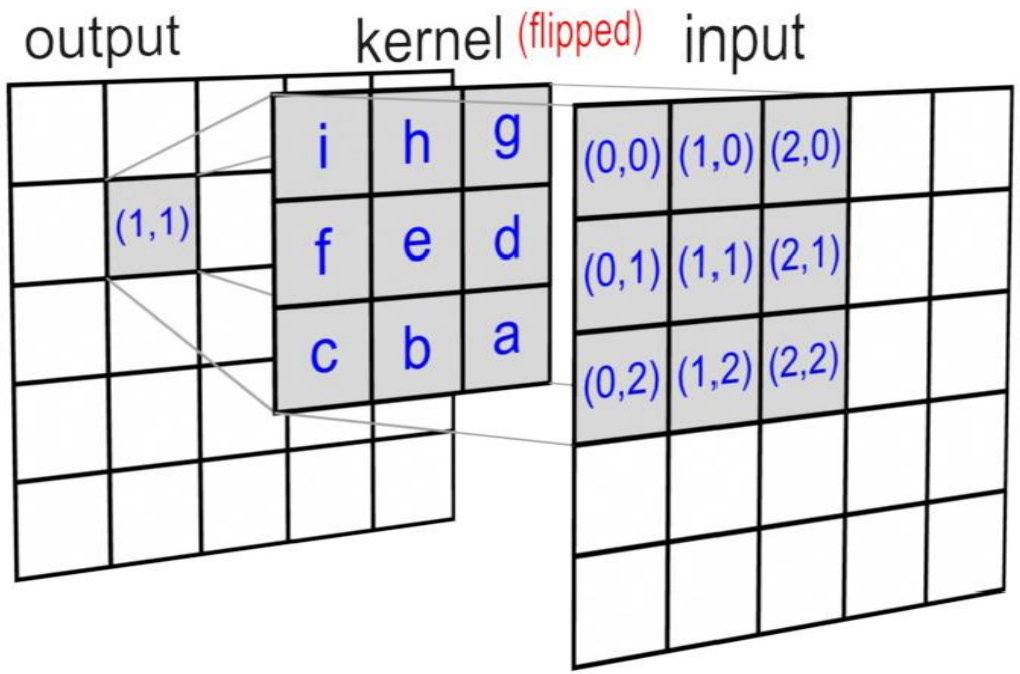


$h[m, n]$

n \ m	-1	0	1
-1	a	b	c
0	d	e	f
1	g	h	i

$$y[1,1] = \sum_{j=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} x[i,j] \cdot h[1-i, 1-j]$$

$$\begin{aligned} = \; & x[0,0] \cdot h[1,1] + x[1,0] \cdot h[0,1] + x[2,0] \cdot h[-1,1] \\ & + x[0,1] \cdot h[1,0] + x[1,1] \cdot h[0,0] + x[2,1] \cdot h[-1,0] \\ & + x[0,2] \cdot h[1,-1] + x[1,2] \cdot h[0,-1] + x[2,2] \cdot h[-1,-1] \end{aligned}$$



$$y[m, n] = x[m, n] * h[m, n]$$

$$= \sum_{j=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} x[i, j] \cdot h[m - i, n - j]$$

$x[m, n]$

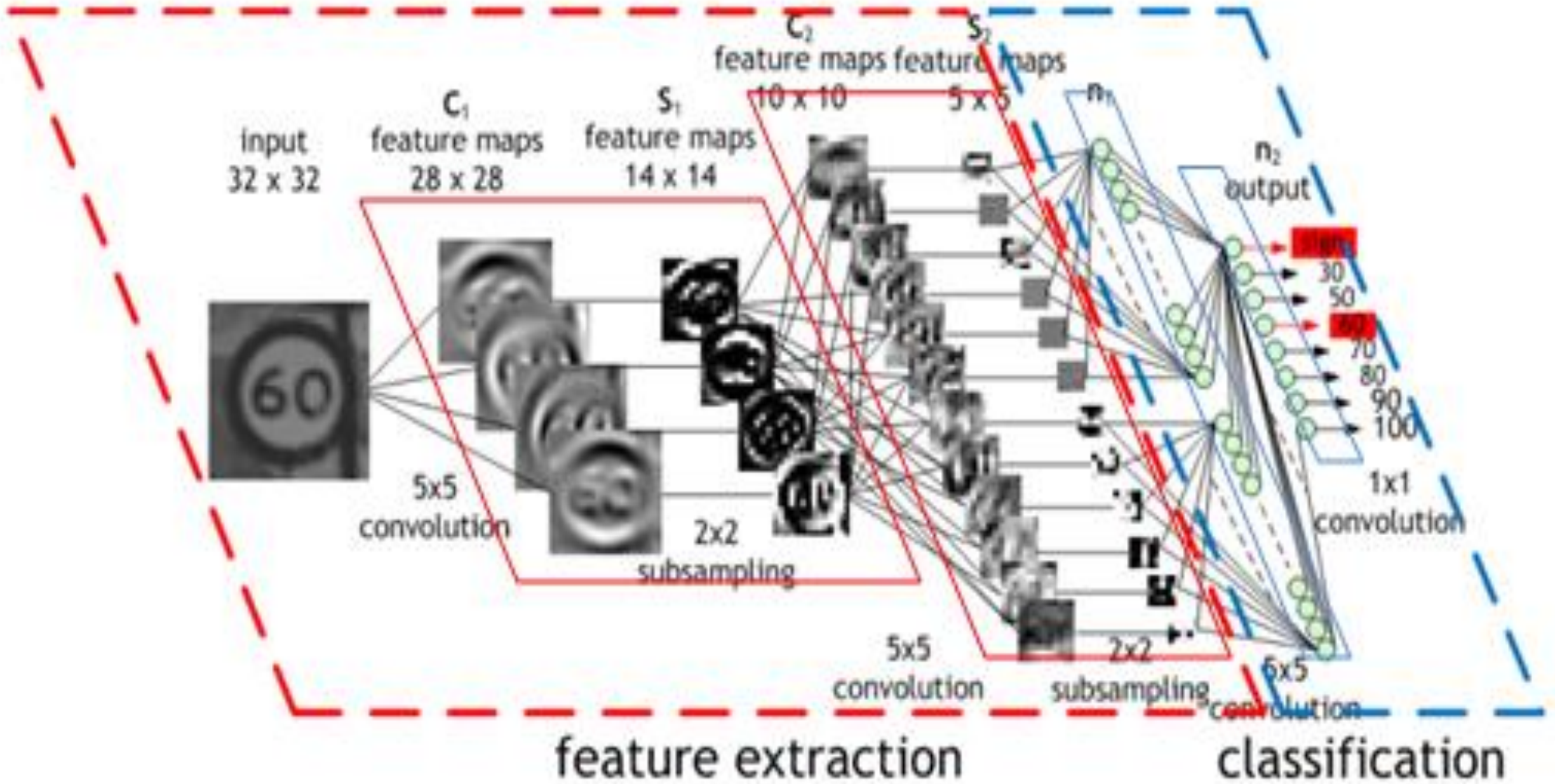
1	2	3
4	5	6
7	8	9

$h[m, n]$

-1	-2	-1
0	0	0
1	2	1

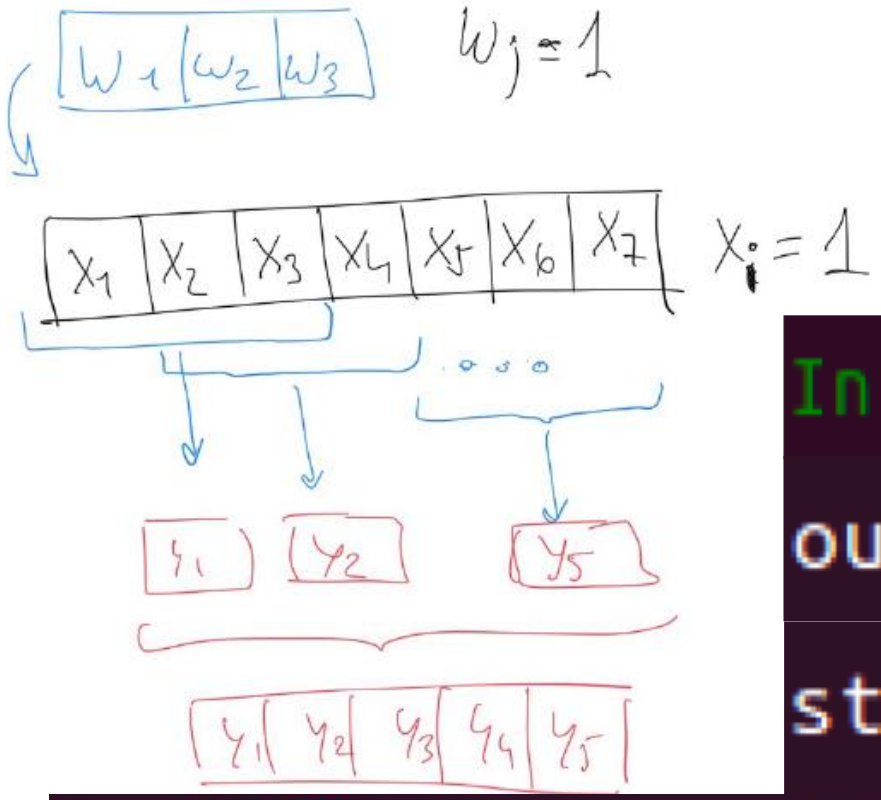
$y[m, n]$

-13	-20	-17
-18	-24	-18
13	20	17



CNN Image by Maurice Peemen

Code: PyTorch Conv1d by Santi Pdp




```
In [1]: import torch
In [2]: import torch.nn as nn
In [4]: x = torch.ones(1, 1, 7)
In [5]: conv = nn.Conv1d(in_channels=1,
out_channels=1, kernel_size=3,
stride=1, padding=0, bias=False)
In [6]: conv.weight.data = torch.ones(1, 1, 3)
In [7]: y = conv(x)  In [8]: y
Out[8]: tensor([[[ 3.,  3.,  3.,  3.,  3.]]])
```