Convolutional Neural Networks

Swati Mishra

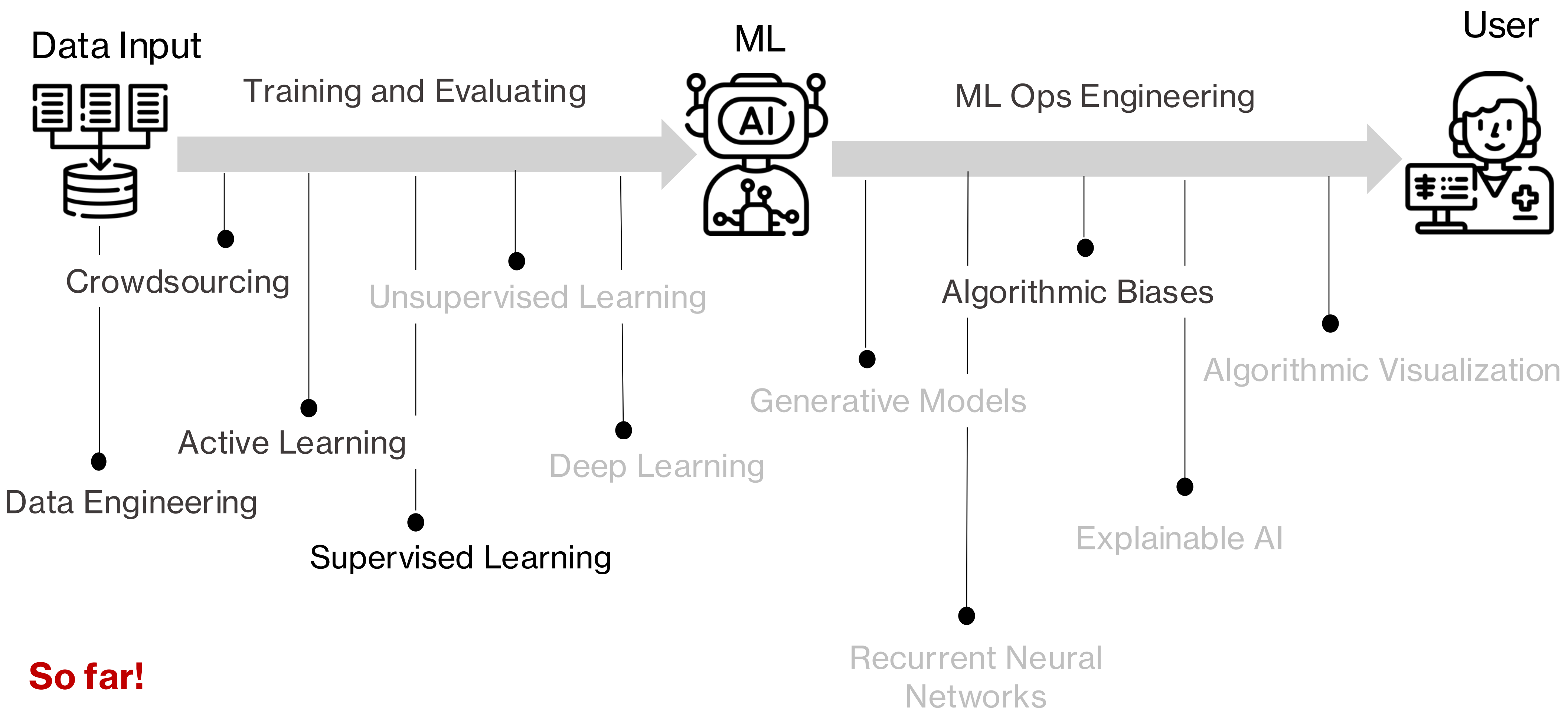
Applications of Machine Learning (4AL3)

Fall 2024

McMaster University

ENGINEERING

Review



Data Input

Training and Evaluating

ML

ML Ops Engineering

User

Crowdsourcing

Data Engineering

Active Learning

Supervised Learning

Unsupervised Learning

Deep Learning

Generative Models

Recurrent Neural Networks

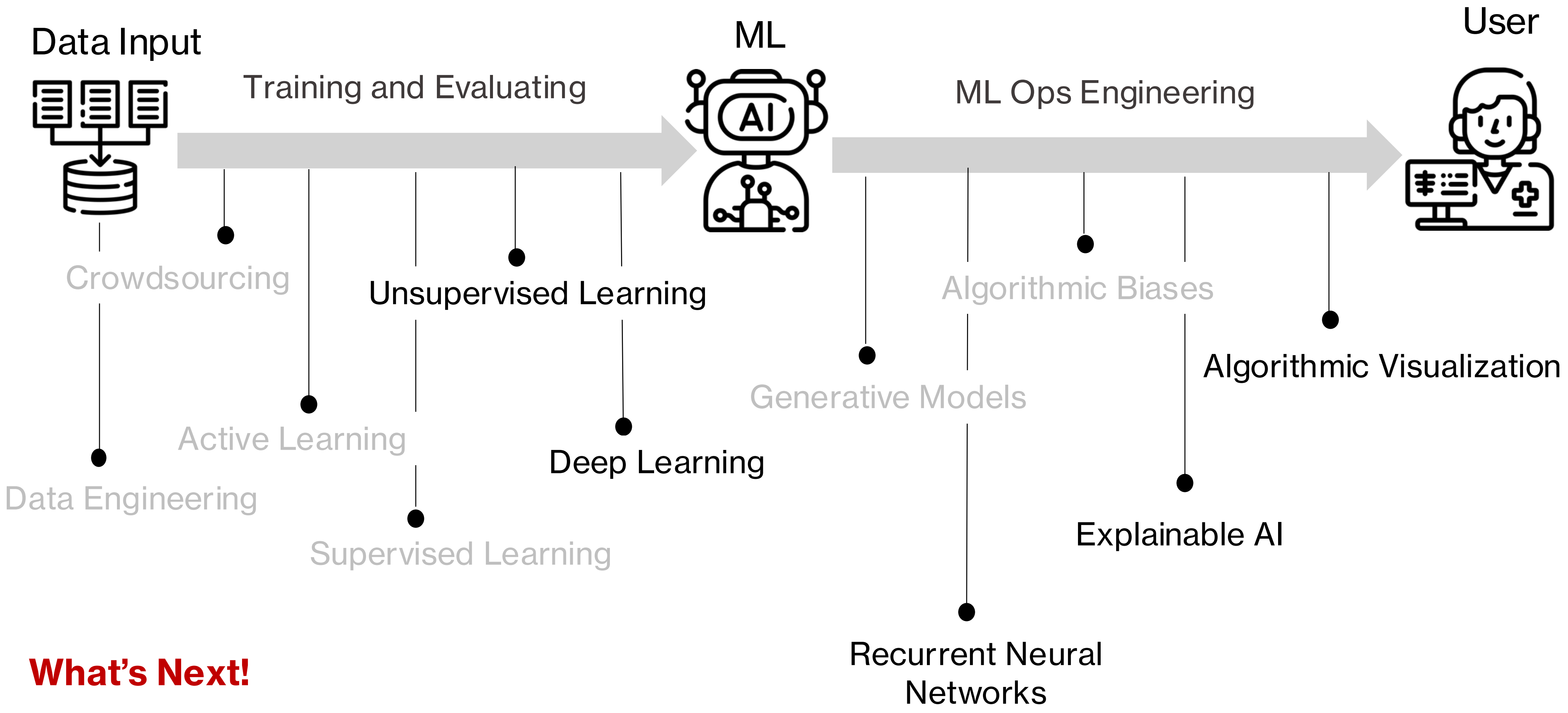
Algorithmic Biases

Explainable AI

Algorithmic Visualization

So far!

Review



Data Input

Training and Evaluating

ML

ML Ops Engineering

User

Crowdsourcing

Data Engineering

Active Learning

Supervised Learning

Unsupervised Learning

Deep Learning

Generative Models

Recurrent Neural Networks

Algorithmic Biases

Explainable AI

Algorithmic Visualization

What's Next!

Convolutional Neural Networks

- Convolutional neural networks are very similar to neural networks that use convolutional operation in place of general matrix multiplication in at least one of the layers.

Convolutional Neural Networks

- Convolutional neural networks are very similar to neural networks that use convolutional operation in place of general matrix multiplication in at least one of the layers.

- Convolution is an operation on two functions of a real valued argument.

$$s(t) = (x * w)(t)$$

Convolutional Neural Networks

- Convolutional neural networks are very similar to neural networks that use convolutional operation in place of general matrix multiplication in at least one of the layers.

- Convolution is an operation on two functions of a real valued argument.

$$\underbrace{s(t)}_{\text{feature map}} = (x * w)(t)$$

- x is an input function
- w is a kernel or filter

Convolutional Neural Networks

- Convolutional neural networks are very similar to neural networks that use convolutional operation in place of general matrix multiplication in at least one of the layers.

- Convolution is an operation on two functions of a real valued argument.

$$\underbrace{s(t)}_{\text{feature map}} = (x * w)(t)$$

- x is an input function
- w is a kernel or filter

Discreet convolution:

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a)$$

Convolutional Neural Networks

- Convolutional neural networks are very similar to neural networks that use convolutional operation in place of general matrix multiplication in at least one of the layers.
- Convolution is an operation on two functions of a real valued argument.

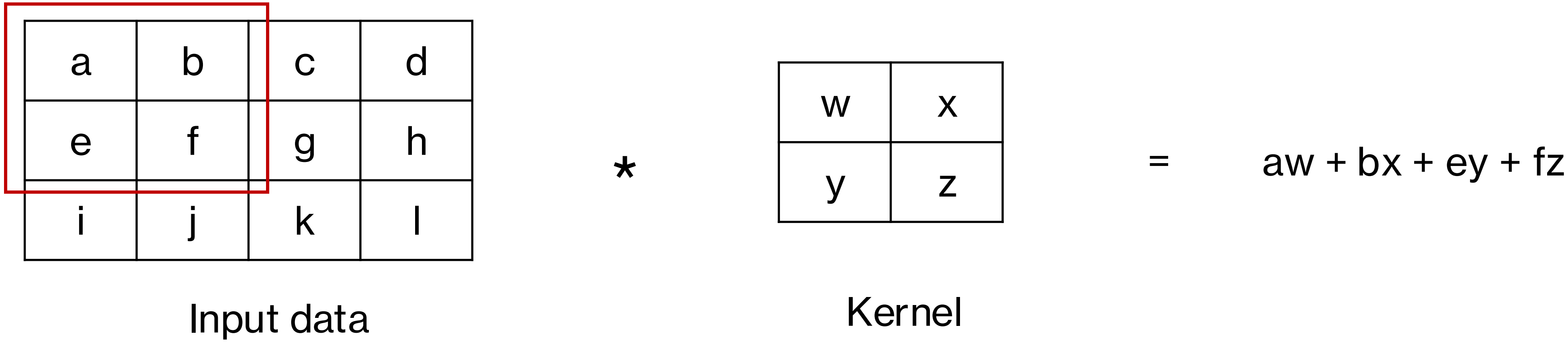
$$\underbrace{s(t)}_{\text{feature map}} = (x * w)(t)$$

- x is an input function
- w is a kernel or filter

Discreet convolution:

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m, n) K(i - m, j - n).$$

Convolutional Operation



a	b	c	d
e	f	g	h
i	j	k	l

Input data

*

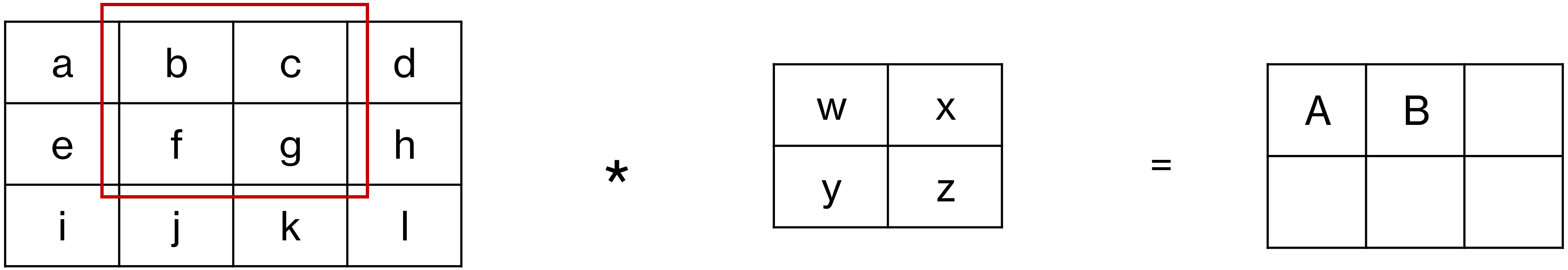
w	x
y	z

Kernel

= aw + bx + ey + fz

popularly called as “filter”

Convolutional Operation

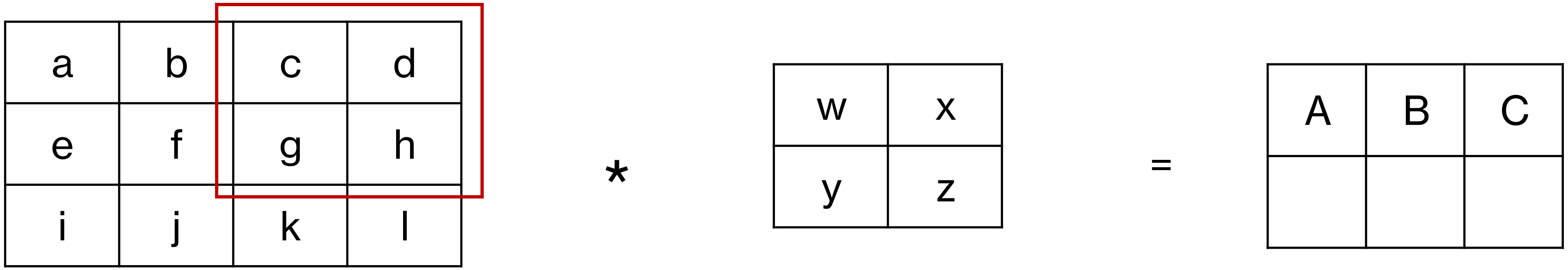


$$
\begin{array}{|c|c|c|c|}
\hline
a & b & c & d \\
\hline
e & f & g & h \\
\hline
i & j & k & l \\
\hline
\end{array}
\quad * \quad
\begin{array}{|c|c|}
\hline
w & x \\
\hline
y & z \\
\hline
\end{array}
\quad = \quad
\begin{array}{|c|c|c|}
\hline
A & B & \\
\hline
 & & \\
\hline
\end{array}
$$

$$A = aw + bx + ey + fz$$

$$B = bw + cx + fy + gz$$

Convolutional Operation



a	b	c	d
e	f	g	h
i	j	k	l

*

w	x
y	z

=

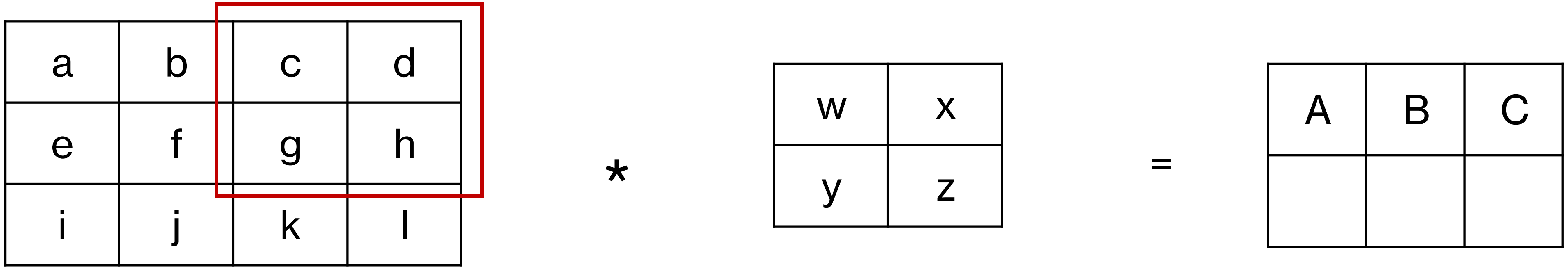
A	B	C

$$A = aw + bx + ey + fz$$

$$B = bw + cx + fy + gz$$

$$C = cw + dx + gy + hz$$

Convolutional Operation



$$\begin{bmatrix}a & b & c & d\\ e & f & g & h\\ i & j & k & l\end{bmatrix} * \begin{bmatrix}w & x\\ y & z\end{bmatrix} = \begin{bmatrix}A & B & C\\ & & \end{bmatrix}$$

$$A = aw + bx + ey + fz$$

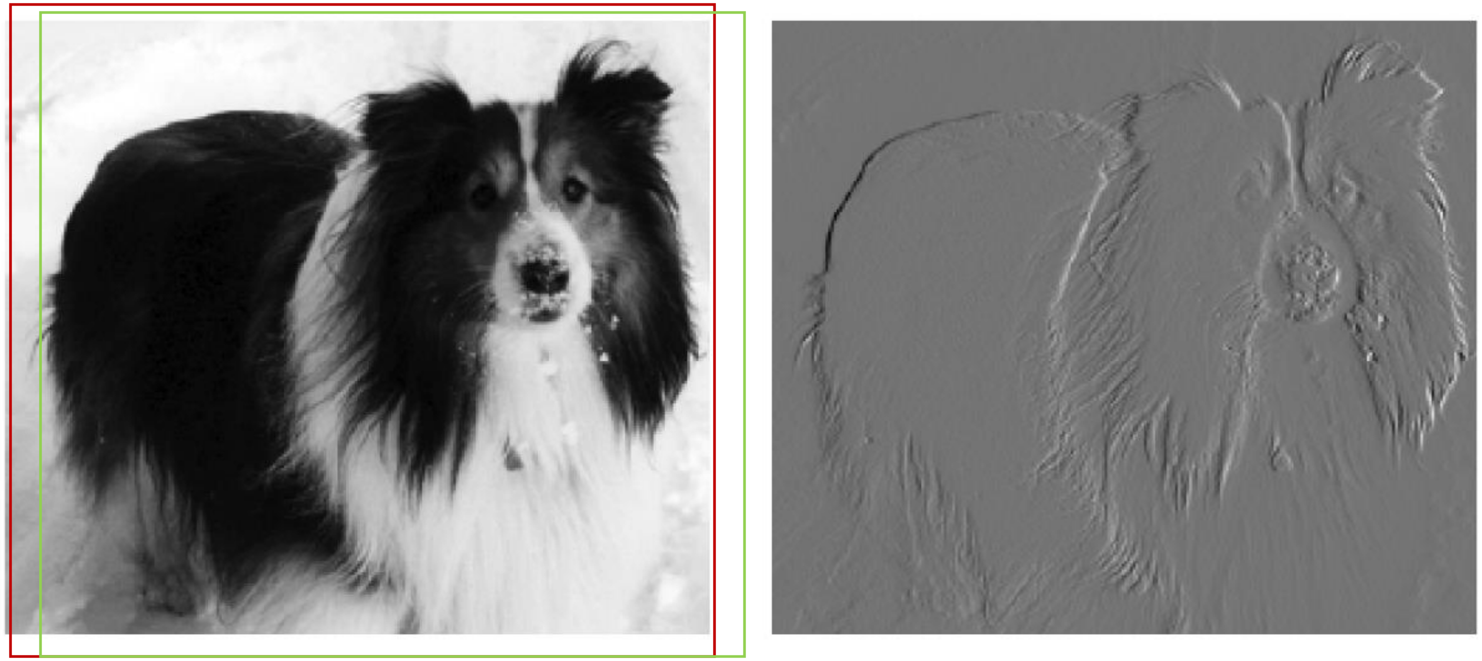
$$B = bw + cx + fy + gz$$

$$C = cw + dx + gy + hz$$

Very efficient for parallelization!

Convolutional Operation: Advantage

- Convolutional operation has an advantage of matrix multiplication.
- The image on the right was formed by taking each pixel in the original image and subtracting the value of its neighboring pixel on the left.



Convolutional Operation

a	b	c	d
e	f	g	h
i	j	k	l

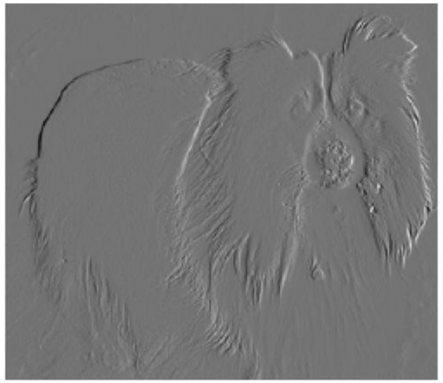
*

w	x
y	z

=

A		

The image on the right was formed by taking each pixel in the original image and subtracting the value of its neighboring pixel on the left.



Operation we want to do: $A = b - a$

$$A = aw + bx + ey + fz$$

Convolutional Operation

a	b	c	d
e	f	g	h
i	j	k	l

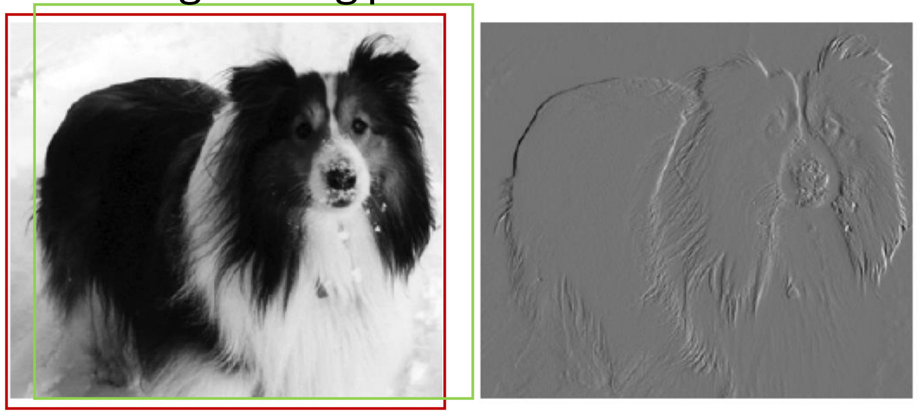
*

1	-1
0	0

=

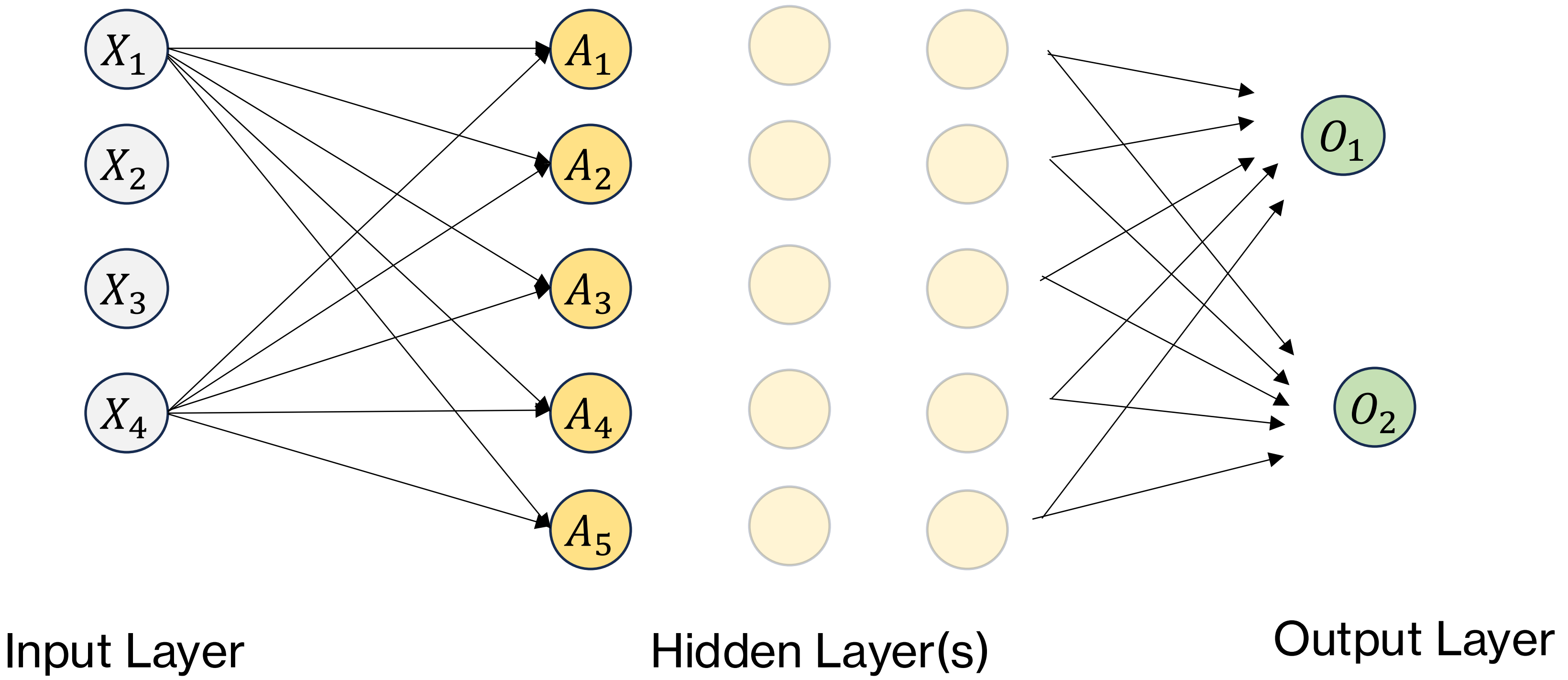
$b - a$		

The image on the right was formed by taking each pixel in the original image and subtracting the value of its neighboring pixel on the left.



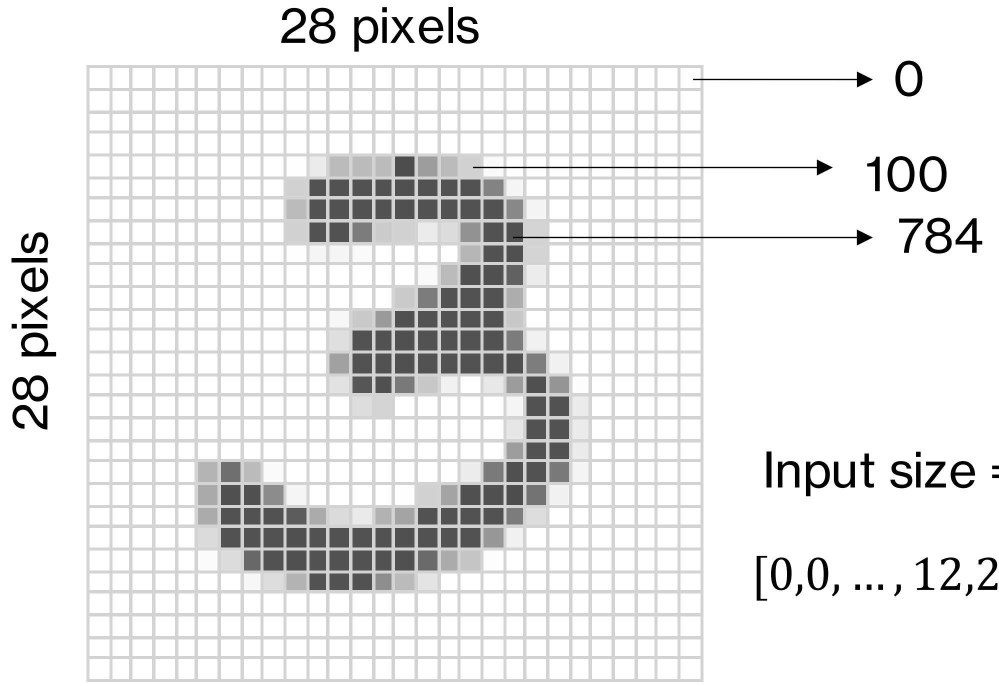
$$A = aw + bx + ey + fz$$

Neural Networks: Architecture



Convolutional NN: Architecture

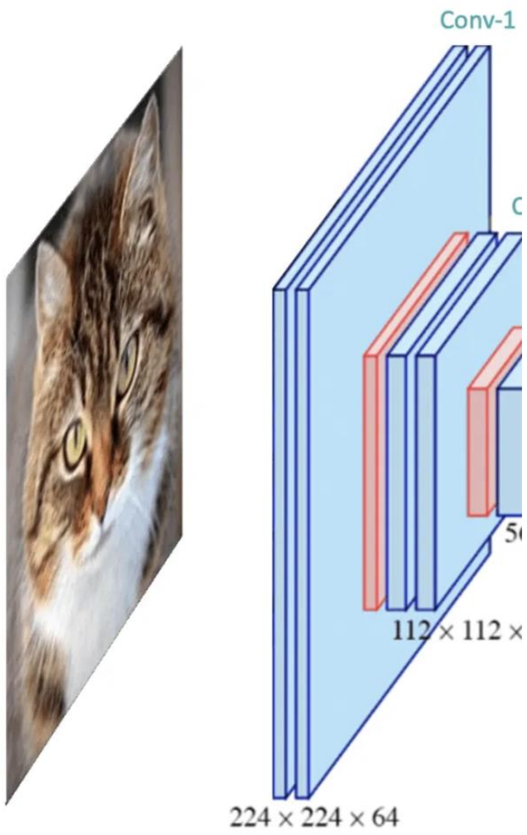
- Transformation of simple image to tensor can be very



Input size = 784

$[0,0, \ldots, 12,23, \ldots 0,0]$

Convolutional Neural Networks



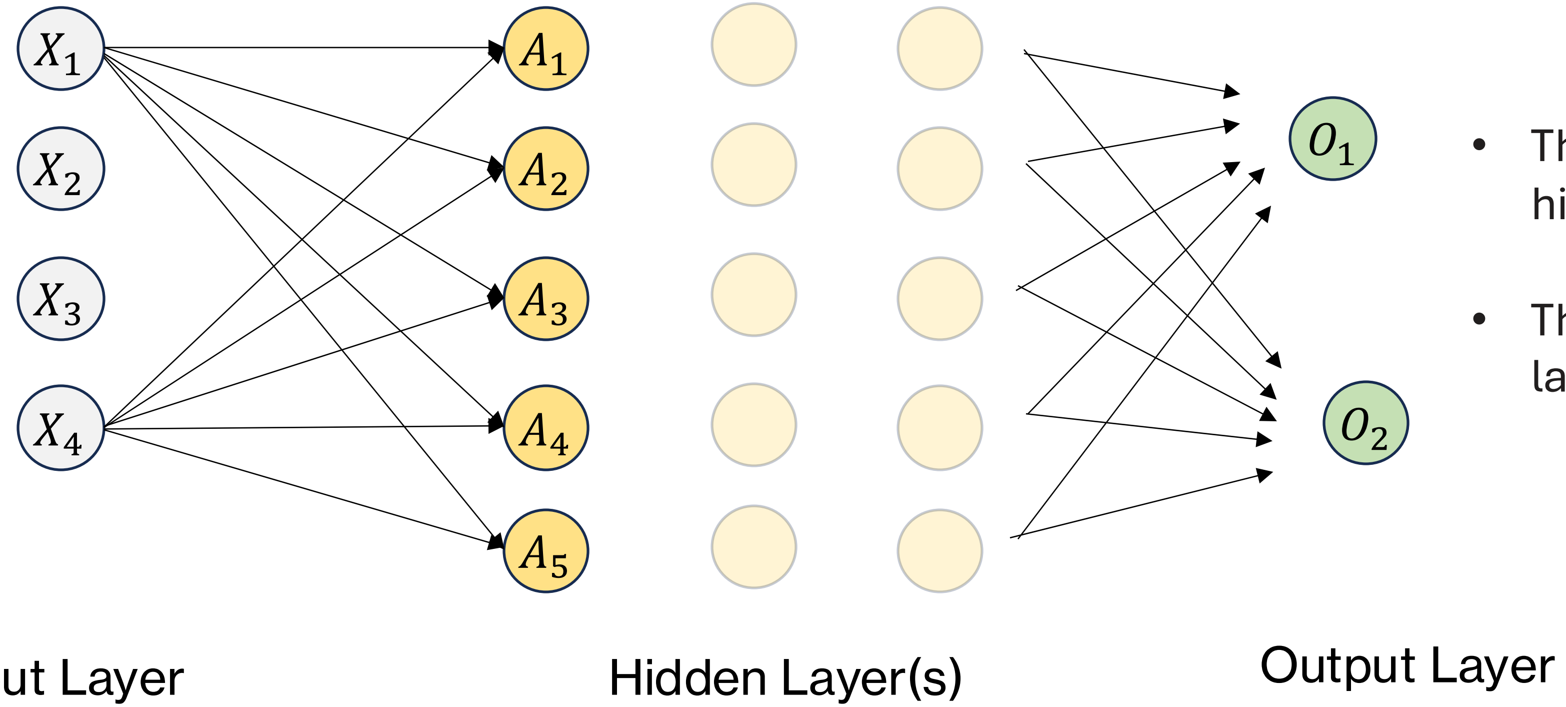
Conv-1:

Height = 224
Width = 224
Depth = 64

Picture Source: https://learnopencv.com/understanding-convolutional-neural-networks-cnn/

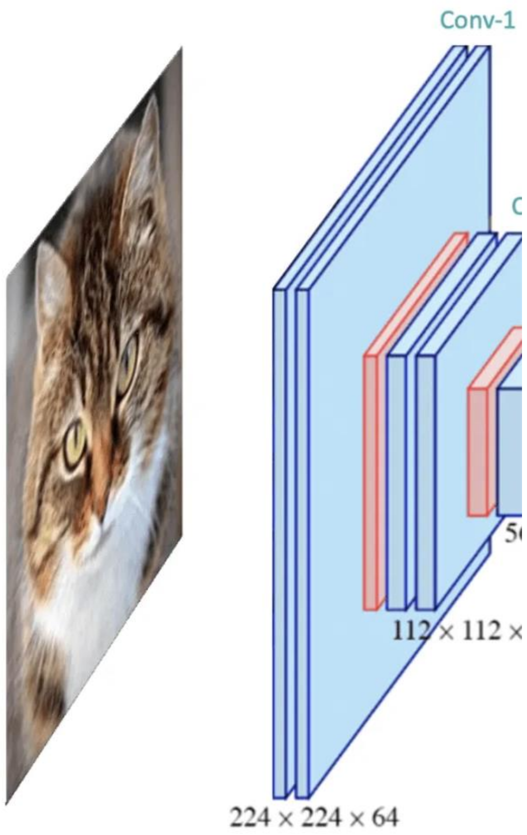
Neural Networks: Architecture

- Depth and width are different for CNNs



- The dimensionality of the hidden layers is called **width**.
- The number of the hidden layers is called the **depth**.

Convolutional Neural Networks



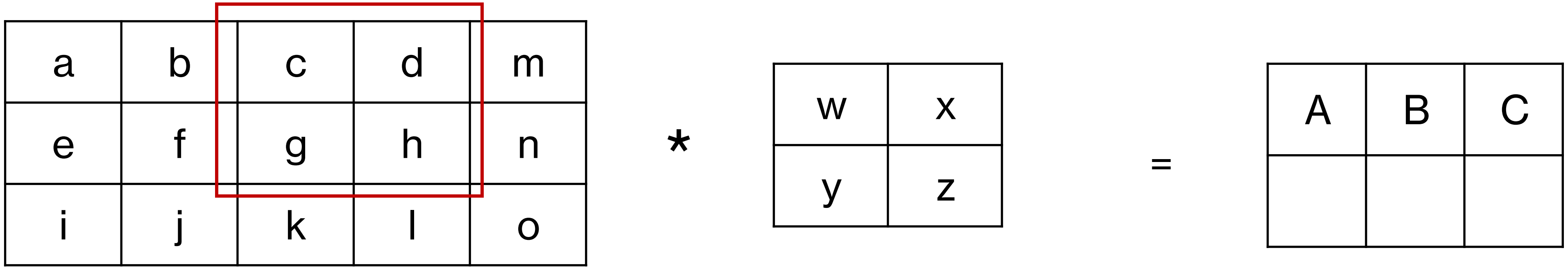
Conv-1:

Height = 224
Width = 224
Depth = 64

This depth is NOT the depth of the network!

Picture Source: https://learnopencv.com/understanding-convolutional-neural-networks-cnn/

Convolutional Operation



a	b	c	d	m
e	f	g	h	n
i	j	k	l	o

*

w	x
y	z

=

A	B	C

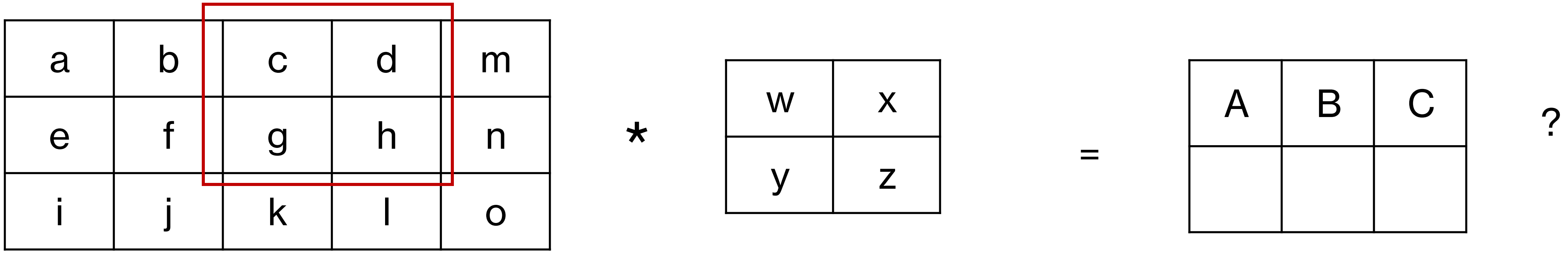
$$A = aw + bx + ey + fz$$

$$B = bw + cx + fy + gz$$

$$C = cw + dx + gy + hz$$

What do we do in this case?

Convolutional Operation



$$\begin{array}{|c|c|c|c|c|}\hline a & b & c & d & m \\ \hline e & f & g & h & n \\ \hline i & j & k & l & o \\ \hline\end{array} * \begin{array}{|c|c|}\hline w & x \\ \hline y & z \\ \hline\end{array} = \begin{array}{|c|c|c|}\hline A & B & C \\ \hline & & \\ \hline\end{array} \quad ?$$

$$A = aw + bx + ey + fz$$

$$B = bw + cx + fy + gz$$

$$C = cw + dx + gy + hz$$

Make the convolution “valid” by not computing last column or use a sliding window operation.

Convolutional Neural Networks

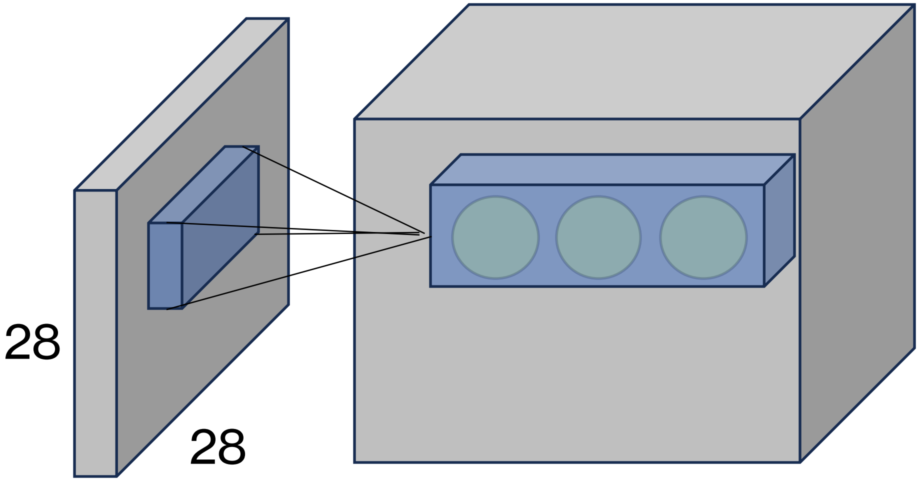
Typically, neural networks have 3 main types of layers:

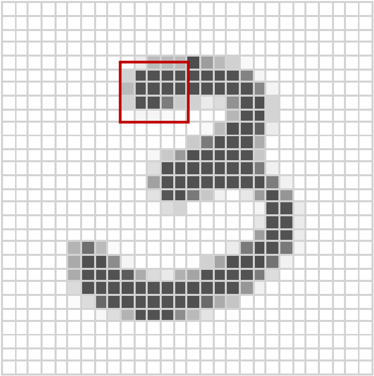
- Convolutional Layer
- Pooling Layer
- Fully-Connected Layer
- Activation function = ReLU Activation

Convolutional Neural Networks

Typically, neural networks have 3 main types of layers:

- Convolutional Layer
- Pooling Layer
- Fully-Connected Layer
- Activation function = ReLU Activation





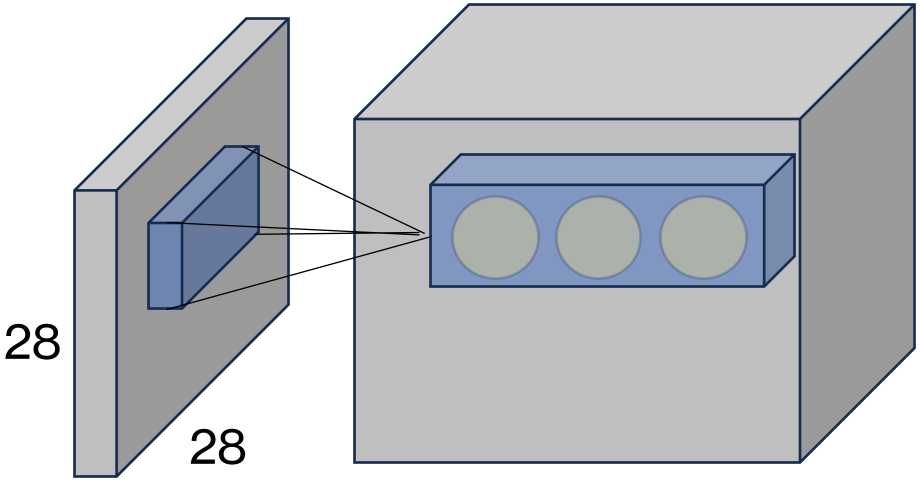
- Filters do not have the same weights
- Filters look at the same region.

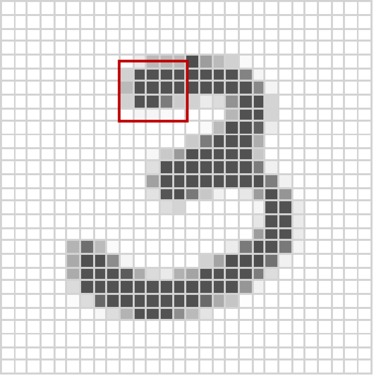
○ = 1 filter

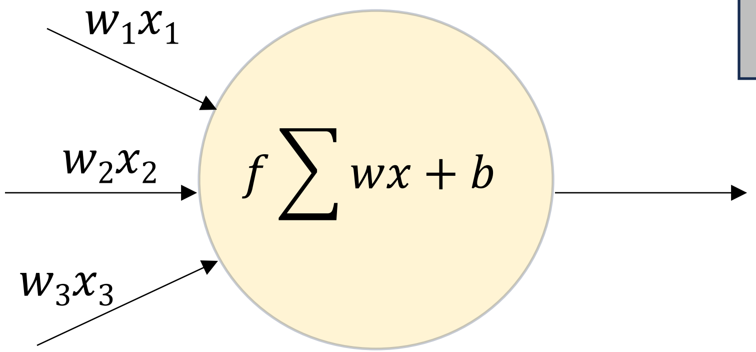
Convolutional Neural Networks

Typically, neural networks have 3 main types of layers:

- Convolutional Layer
- Pooling Layer
- Fully-Connected Layer
- Activation function = ReLU Activation



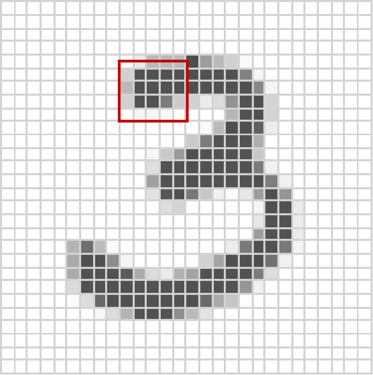


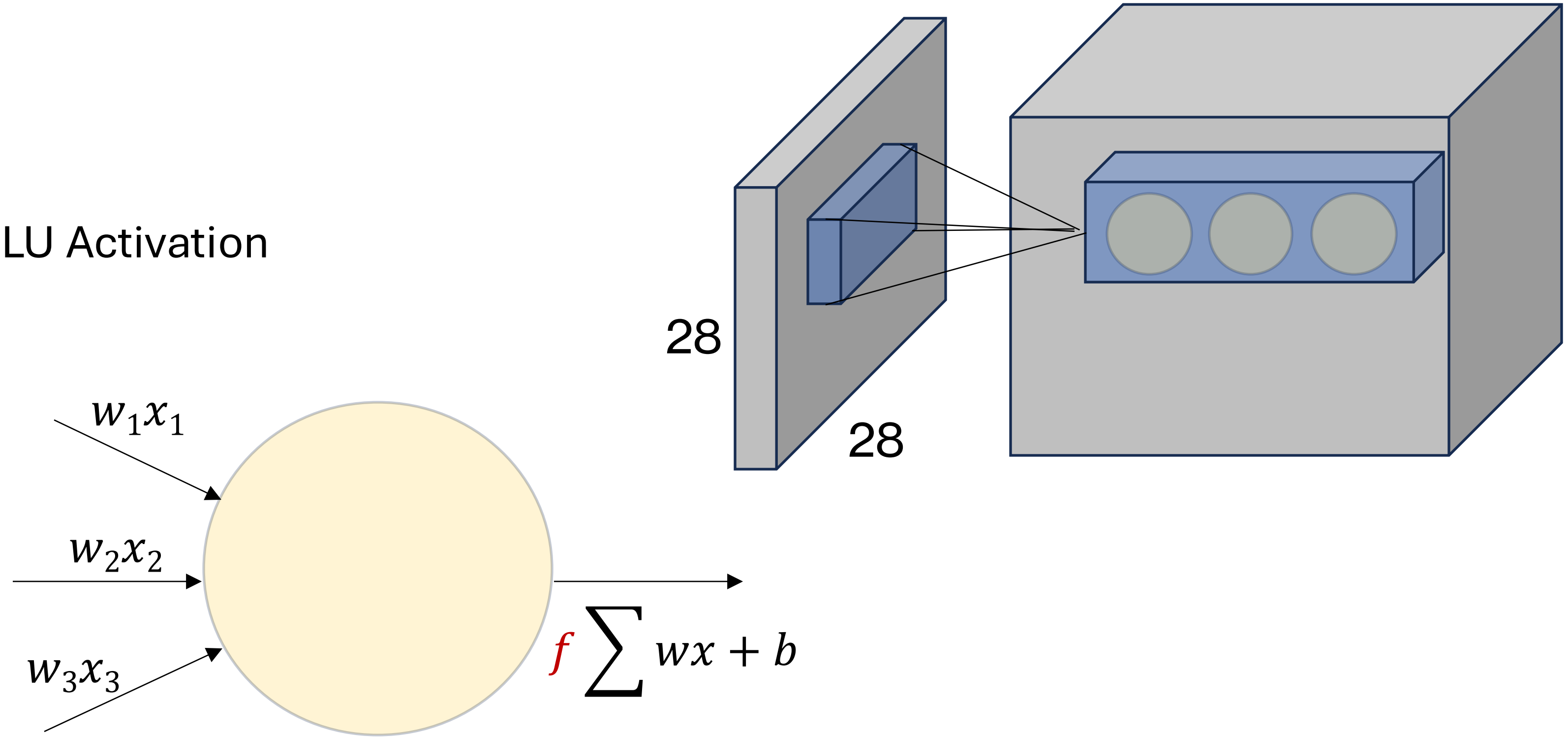


Convolutional Neural Networks

Typically, neural networks have 3 main types of layers:

- Convolutional Layer
- Pooling Layer
- Fully-Connected Layer
- Activation function = ReLU Activation



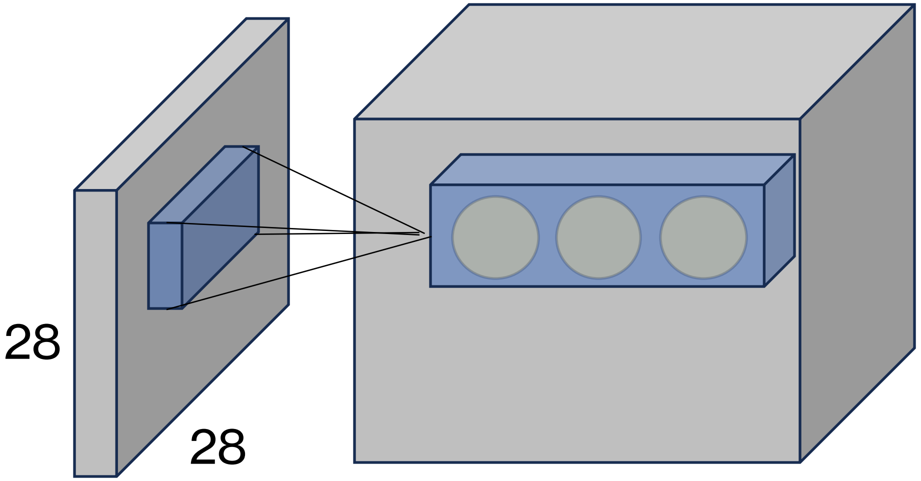


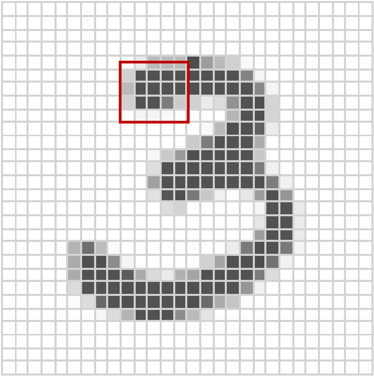
Convolutional Neural Networks

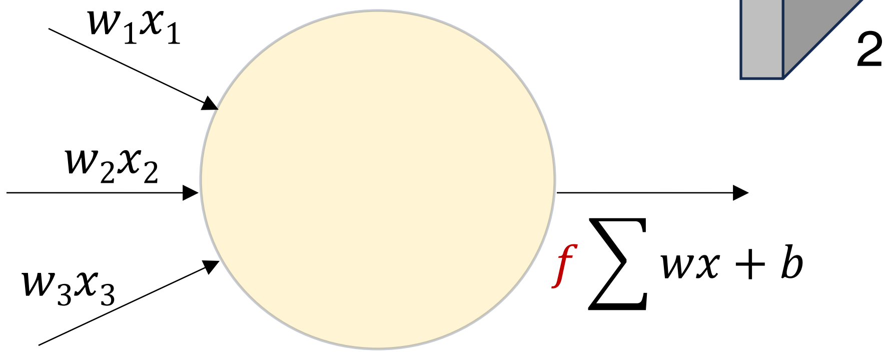
Typically, neural networks have 3 main types of layers:

- Convolutional Layer
- Pooling Layer
- Fully-Connected Layer
- Activation function = ReLU Activation

Layer stacking order: INPUT - CONV





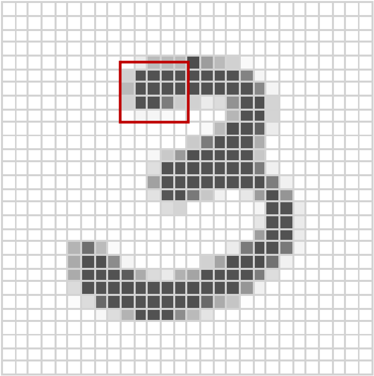


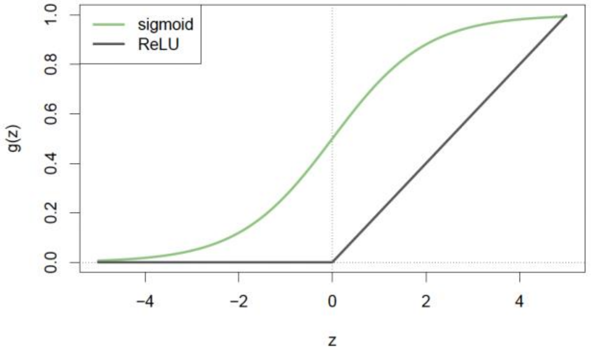
For 3 filters the size of convolutional layer = 28x28x3

Convolutional Neural Networks

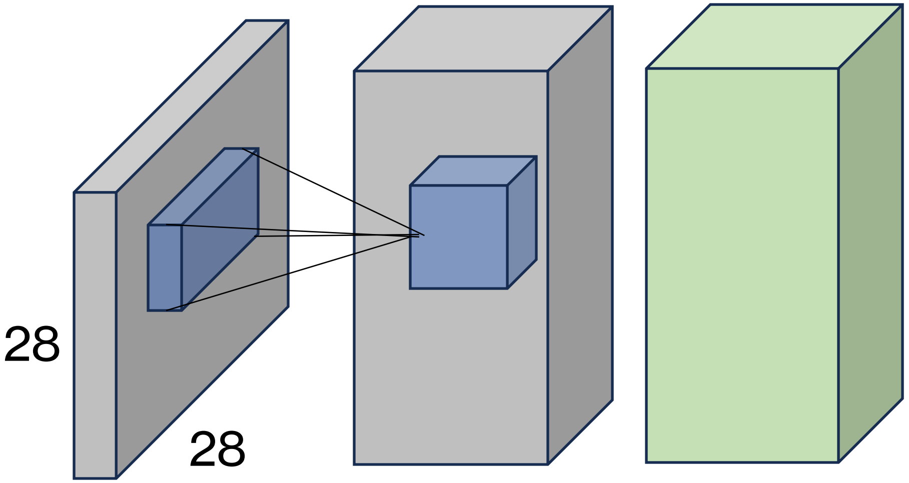
Typically, neural networks have 3 main types of layers:

- Convolutional Layer
- Pooling Layer
- Fully-Connected Layer
- ReLU Layer – element wise





Layer stacking order: INPUT – CONV -RELU



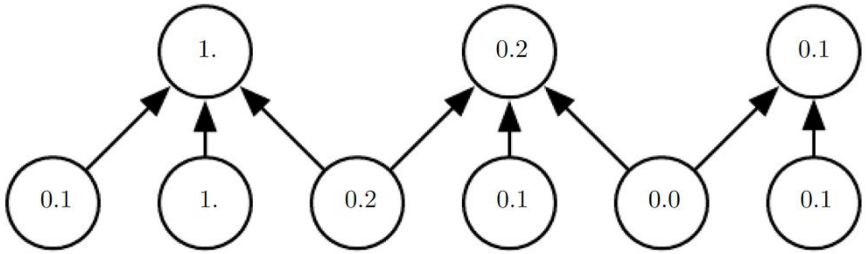
For 3 filters the size of convolutional layer = 28x28x3

Output of ReLU Layer ?

Convolutional Neural Networks

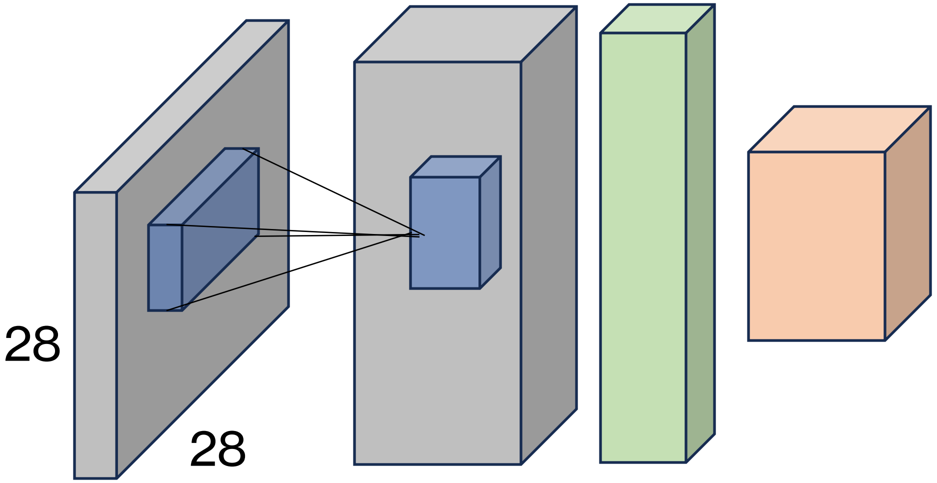
Typically, neural networks have 3 main types of layers:

- Convolutional Layer
- Pooling Layer
- Fully-Connected Layer
- ReLU Layer – element wise



Max Pooling operation

Layer stacking order: INPUT – CONV –RELU-POOL

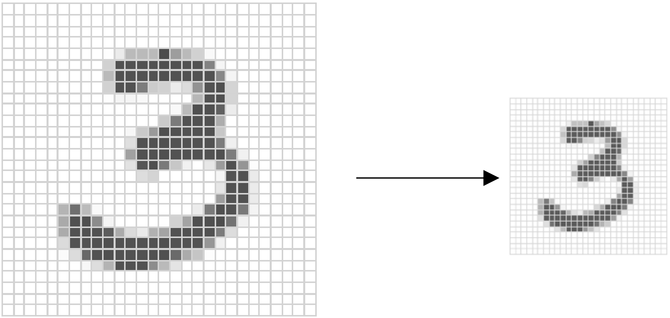


Pooling layer: down samples the input along height and width.

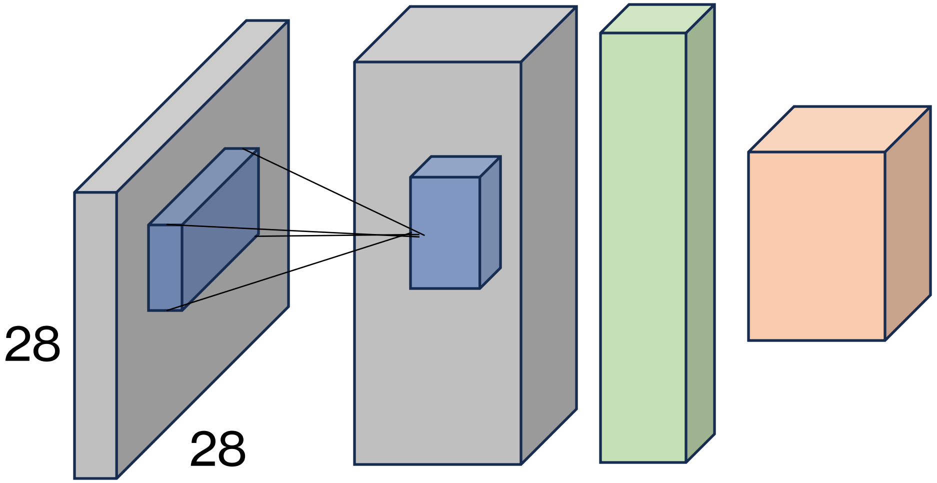
Convolutional Neural Networks

Typically, neural networks have 3 main types of layers:

- Convolutional Layer
- Pooling Layer
- Fully-Connected Layer
- ReLU Layer – element wise



Layer stacking order: INPUT – CONV –RELU-POOL

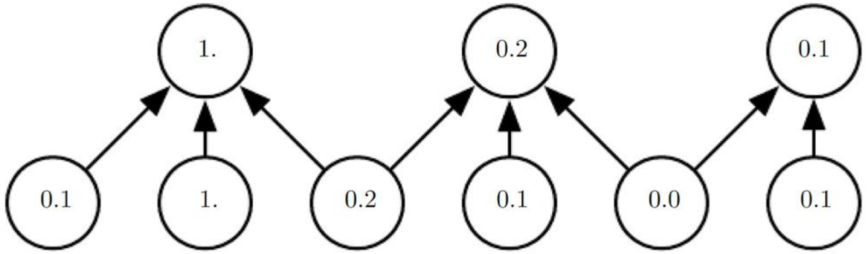


Pooling layer: down samples the input along height and width. Also called **detector**.

Convolutional Neural Networks

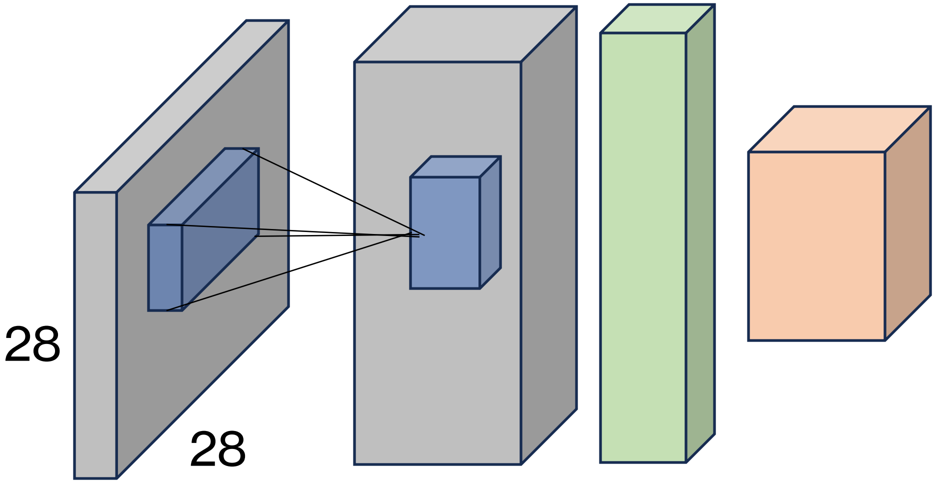
Typically, neural networks have 3 main types of layers:

- Convolutional Layer
- Pooling Layer
- Fully-Connected Layer
- ReLU Layer – element wise



Max Pooling operation

Layer stacking order: INPUT – CONV –RELU-POOL



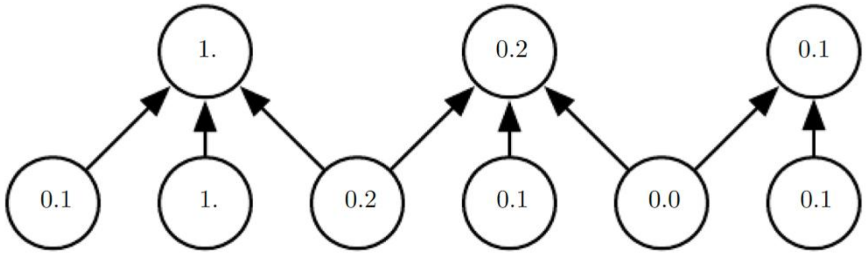
Pooling layer: down samples the input along height and width.

Output of Pooling Layer ?

Convolutional Neural Networks

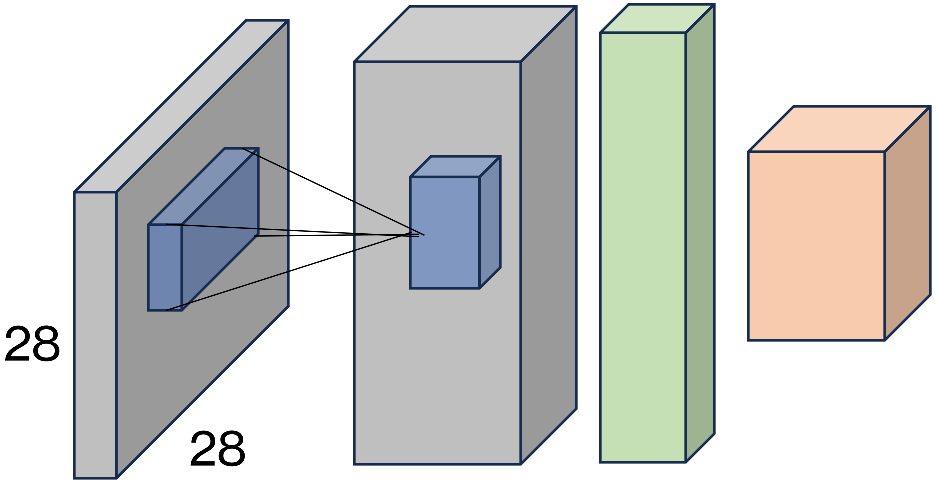
Typically, neural networks have 3 main types of layers:

- Convolutional Layer
- Pooling Layer
- Fully-Connected Layer
- ReLU Layer – element wise



Max Pooling operation

Layer stacking order: INPUT – CONV –RELU-POOL



Pooling layer: down samples the input along height and width.

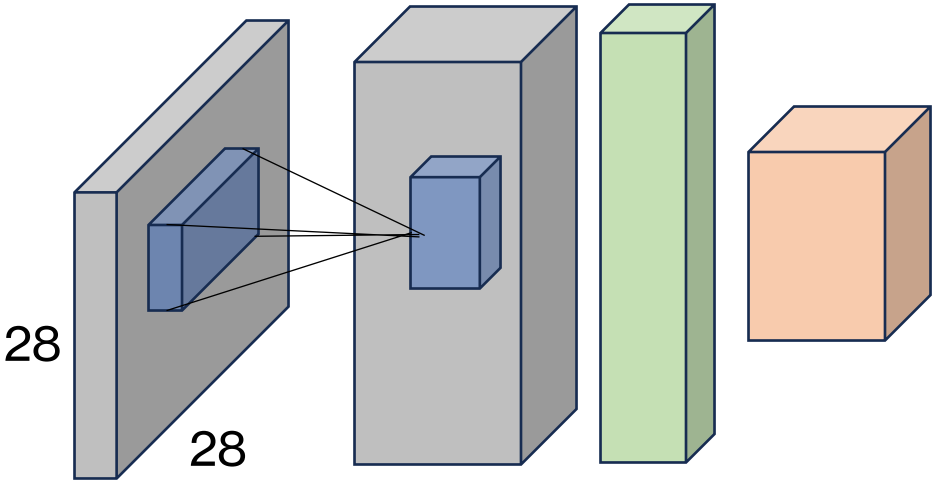
Output of Pooling Layer = 16x16x3

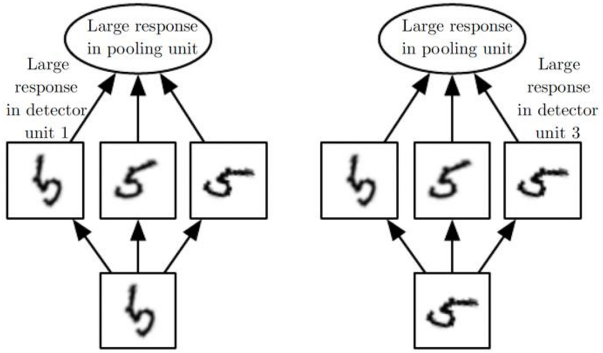
Convolutional Neural Networks

Typically, neural networks have 3 main types of layers:

- Convolutional Layer
- Pooling Layer
- Fully-Connected Layer
- ReLU Layer – element wise

Layer stacking order: INPUT – CONV –RELU-POOL




Large response in pooling unit
Large response in detector unit 1
Large response in pooling unit
Large response in detector unit 3

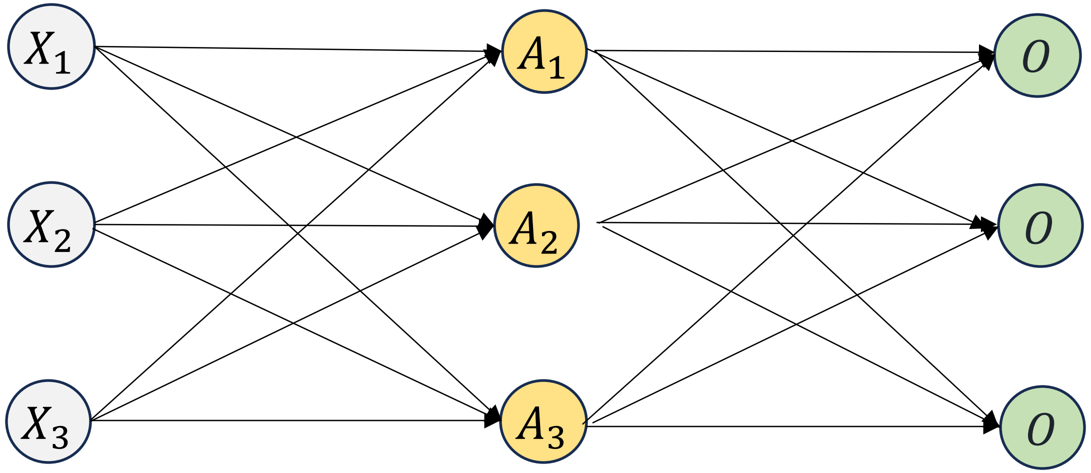
Max Pooling operation

A pooling unit that pools over multiple features that are learned with separate parameters can learn to be invariant to transformations of the input.

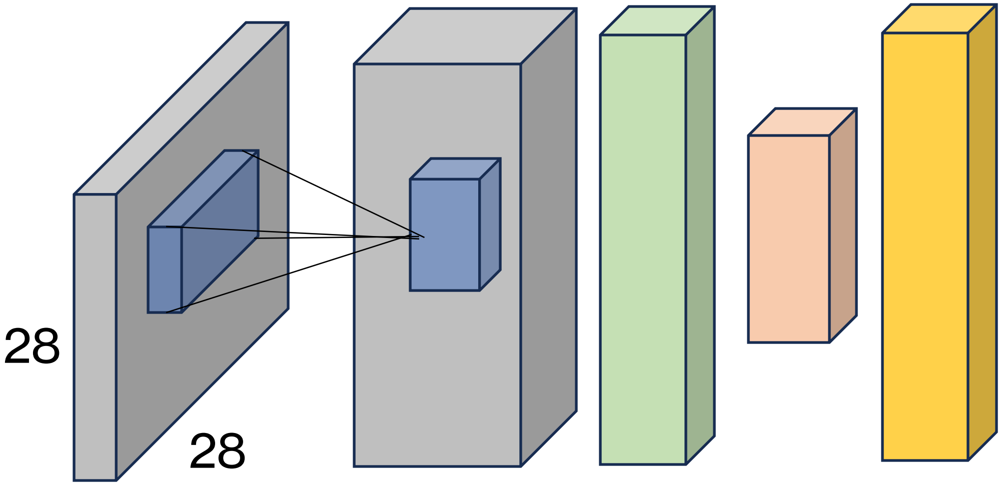
Convolutional Neural Networks

Typically, neural networks have 3 main types of layers:

- Convolutional Layer
- Pooling Layer
- Fully-Connected Layer
- ReLU Layer – element wise



Layer stacking order: CONV – RELU –POOL-FC

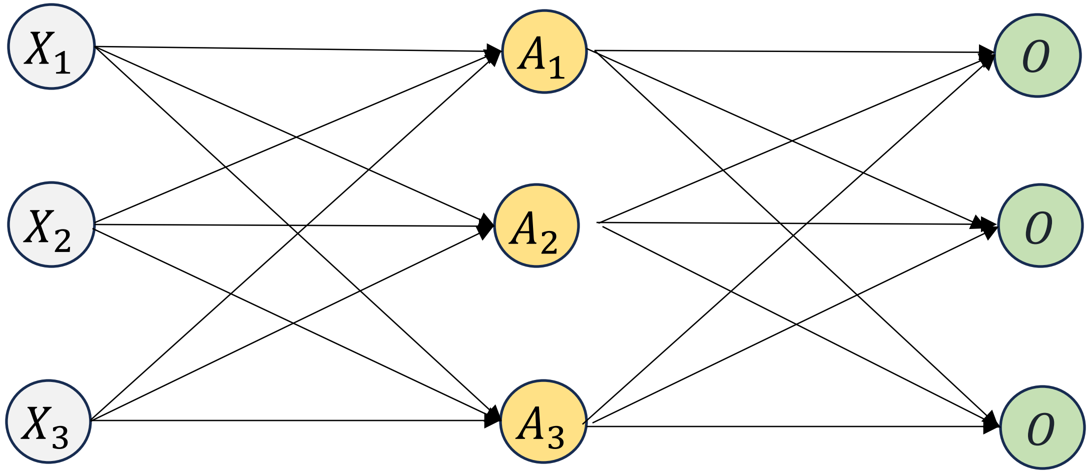


Fully connected layer computes class scores and is like just another neural network.

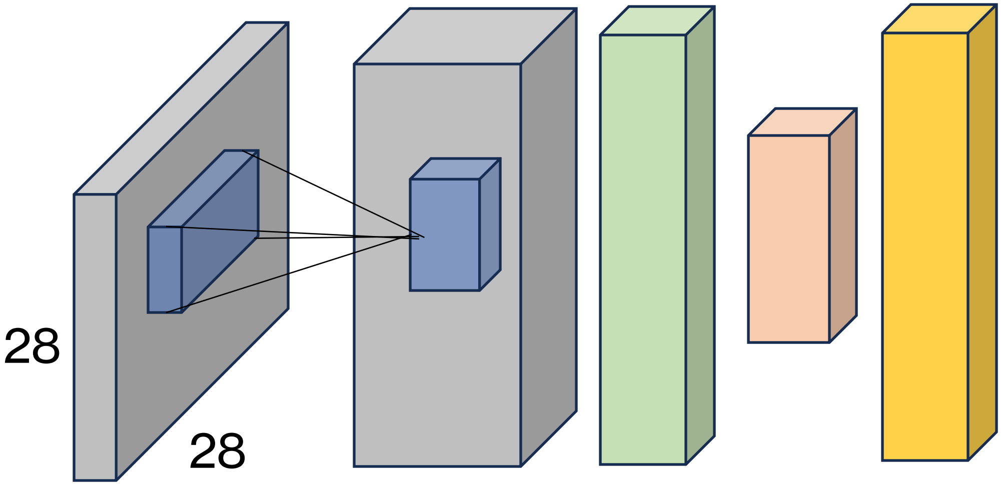
Convolutional Neural Networks

Typically, neural networks have 3 main types of layers:

- Convolutional Layer
- Pooling Layer
- Fully-Connected Layer
- ReLU Layer – element wise



Layer stacking order: CONV – RELU –POOL-FC



Output of fully connected layer: 1x1xN

N of classes = 10 digits

Convolutional Neural Networks

Typically, neural networks have 3 main types of layers:

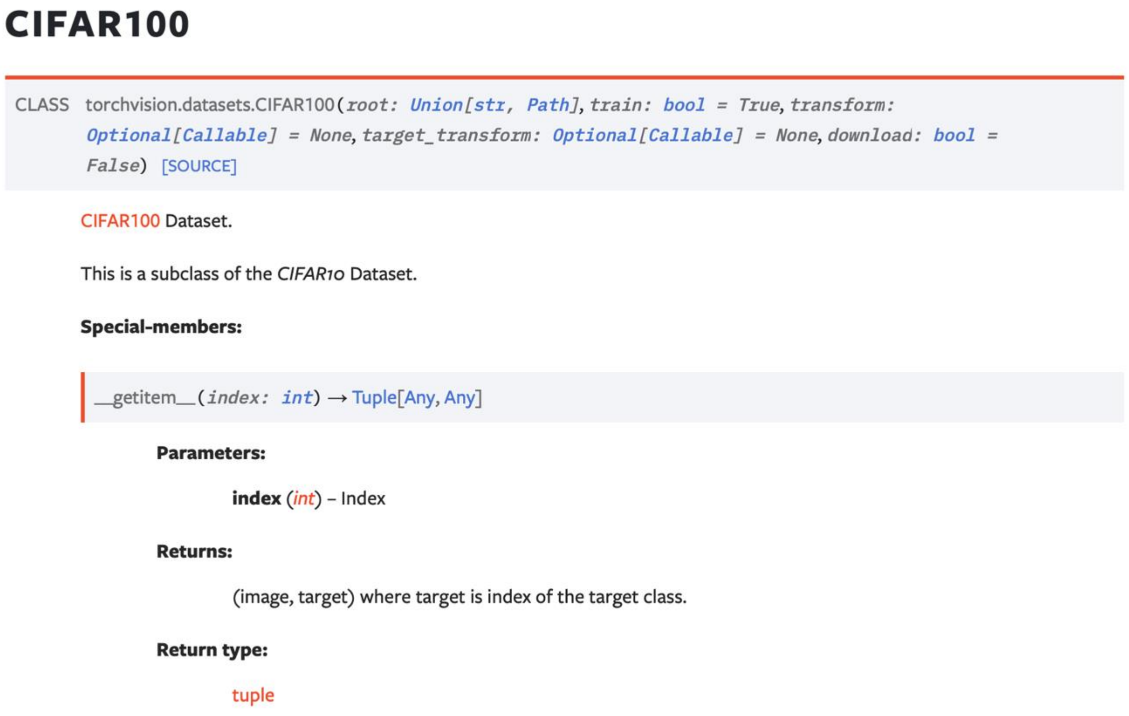
- Convolutional Layer
- Pooling Layer
- Fully-Connected Layer
- RelU activation Layer

Popular CIFAR-100 architecture:

[INPUT - CONV - RELU - POOL - FC]

[INPUT - CONV - RELU - POOL - FC]

⋮



CIFAR100

```
CLASS  torchvision.datasets.CIFAR100(root: Union[str, Path], train: bool = True, transform:
       Optional[Callable] = None, target_transform: Optional[Callable] = None, download: bool =
       False)  [SOURCE]
```

CIFAR100 Dataset.

This is a subclass of the *CIFAR10* Dataset.

Special-members:

```
__getitem__(index: int) → Tuple[Any, Any]
```

Parameters:

index (*int*) – Index

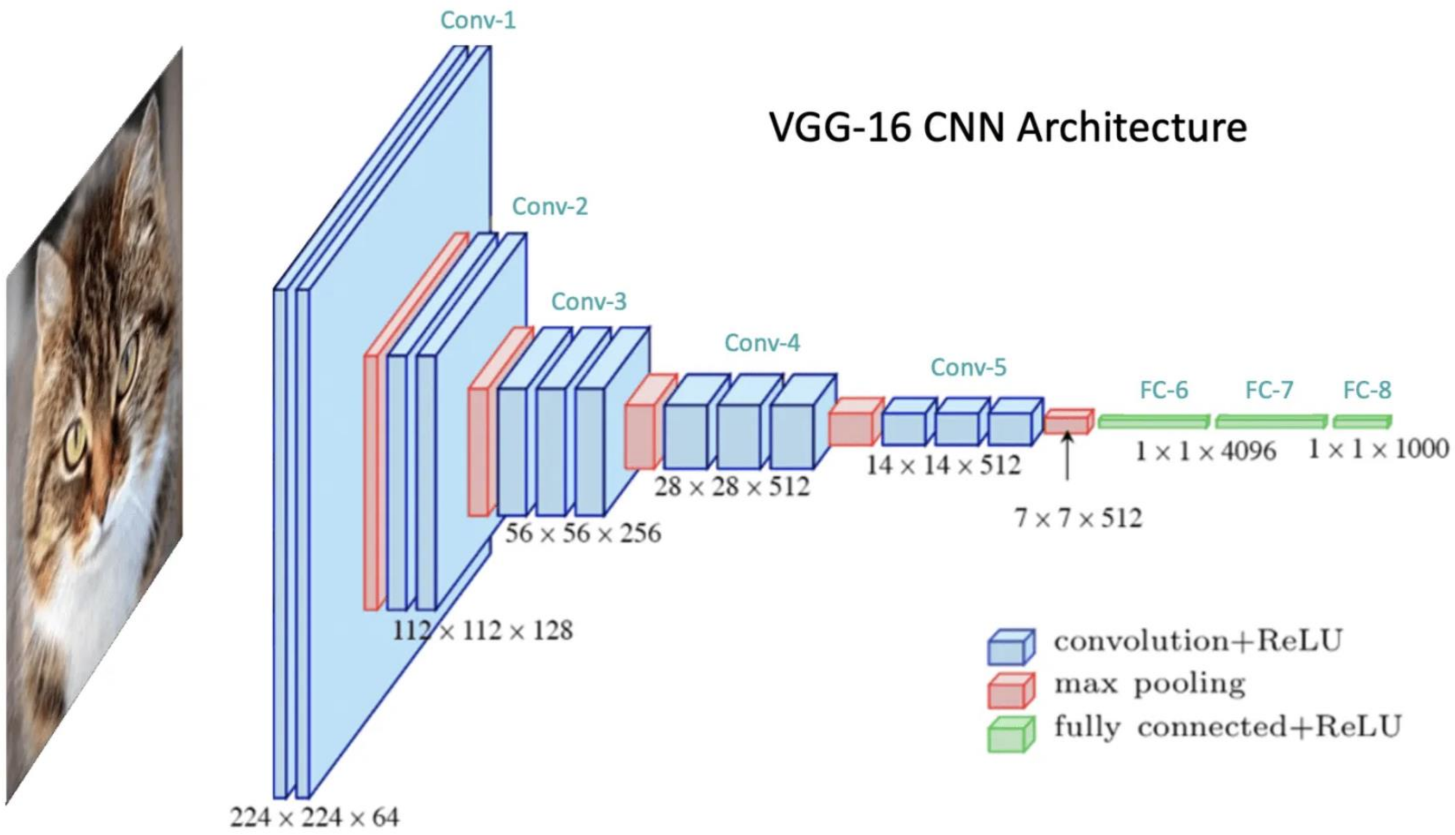
Returns:

(image, target) where target is index of the target class.

Return type:

tuple

Convolutional Neural Networks



Picture Source: https://learnopencv.com/understanding-convolutional-neural-networks-cnn/

Readings

Required Readings:

Introduction to Statistical Learning

- Chapter 10 – Section 10.3 page 406 - 412

***Supplemental Readings (Not required but recommended)*:**

Deep Learning

- Chapter 9 – page 330 - 340

Thank You

McMaster
University