ARTIFICIAL INTELLIGENCE

Presented by Afreen Banu & Ashwin R G

What is AI?

- Artificial Intelligence is composed of two words **Artificial** and **Intelligence**, where Artificial defines *"man-made,"* and intelligence defines *"thinking power"*, hence AI means *"a man-made thinking power.“*
- "It is a branch of computer science by which we can create intelligent machines which can behave like a human, think like humans, and able to make decisions."
- With Artificial Intelligence you do not need to preprogram a machine to do some work, despite that you can create a machine with programmed algorithms which can work with own intelligence, and that is the awesomeness of AI.
- It is believed that AI is not a new technology, and some people says that as per Greek myth, there were Mechanical men in early days which can work and behave like humans.

Why Artificial Intelligence? Necessity of learning AI:

- With the help of AI, you can create such software or devices which can solve real-world problems very easily and with accuracy such as health issues, marketing, traffic issues, etc.
- With the help of AI, you can create your personal virtual Assistant, such as Cortana, Google Assistant, Siri, etc.
- With the help of AI, you can build such Robots which can work in an environment where survival of humans can be at risk.
- AI opens a path for other new technologies, new devices, and new Opportunities.

Goals of Artificial Intelligence

1. Replicate human intelligence
2. Solve Knowledge-intensive tasks
3. An intelligent connection of perception and action
4. Building a machine which can perform tasks that requires human intelligence such as:
    1. Proving a theorem
    2. Playing chess
    3. Plan some surgical operation
    4. Driving a car in traffic
5. Creating some system which can exhibit intelligent behavior, learn new things by itself, demonstrate, explain, and can advise to its user.

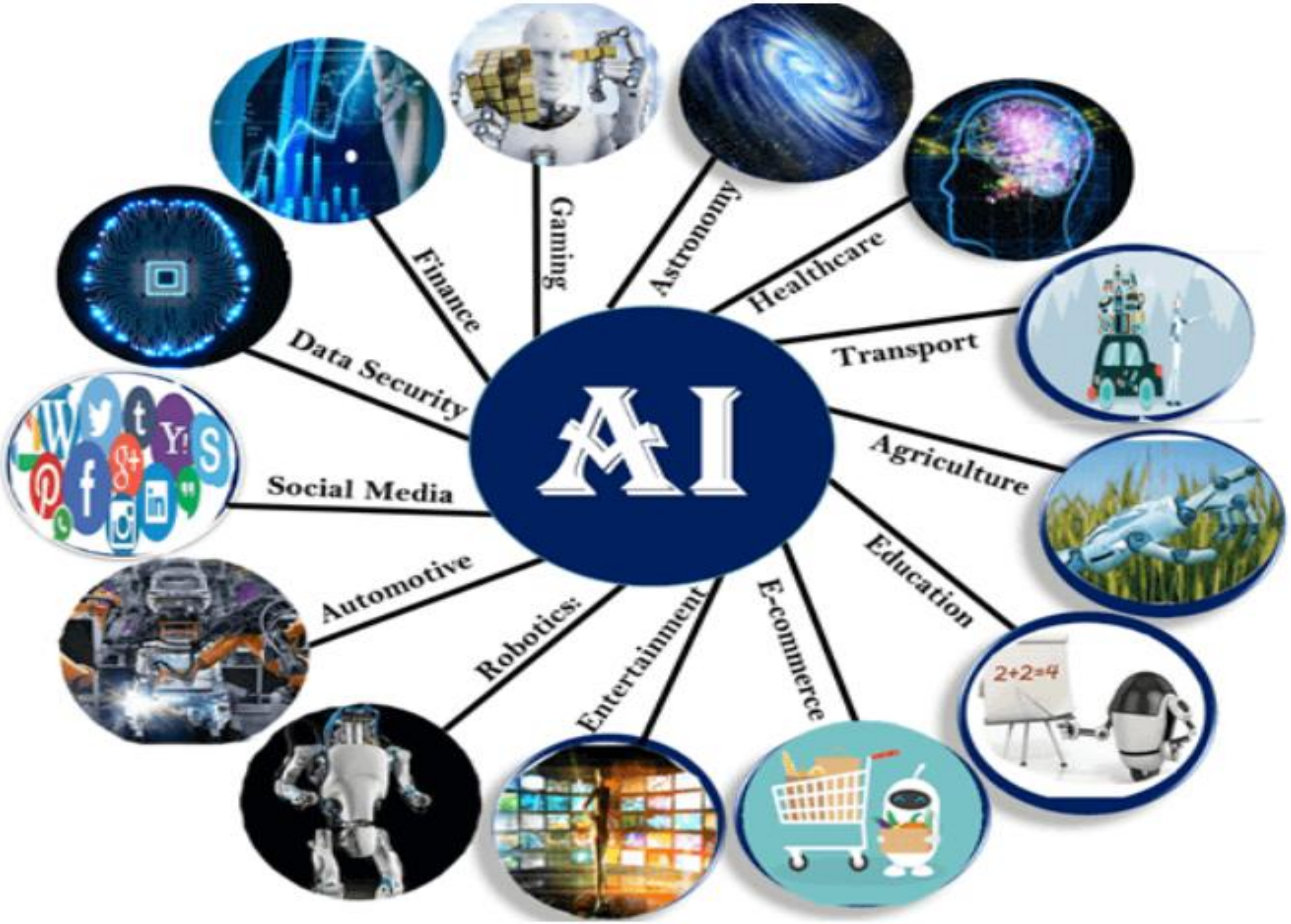
Advantages of Artificial Intelligence

- **High Accuracy with less errors:** AI machines or systems are prone to less errors and high accuracy as it takes decisions as per pre-experience or information.
- **High-Speed:** AI systems can be of very high-speed and fast-decision making, because of that AI systems can beat a chess champion in the Chess game.
- **High reliability:** AI machines are highly reliable and can perform the same action multiple times with high accuracy.
- **Useful for risky areas:** AI machines can be helpful in situations such as defusing a bomb, exploring the ocean floor, where to employ a human can be risky.
- **Digital Assistant:** AI can be very useful to provide digital assistant to the users such as AI technology is currently used by various E-commerce websites to show the products as per customer requirement.
- **Useful as a public utility:** AI can be very useful for public utilities such as a self-driving car which can make our journey safer and hassle-free, facial recognition for security purpose, Natural language processing to communicate with the human in human-language, etc.

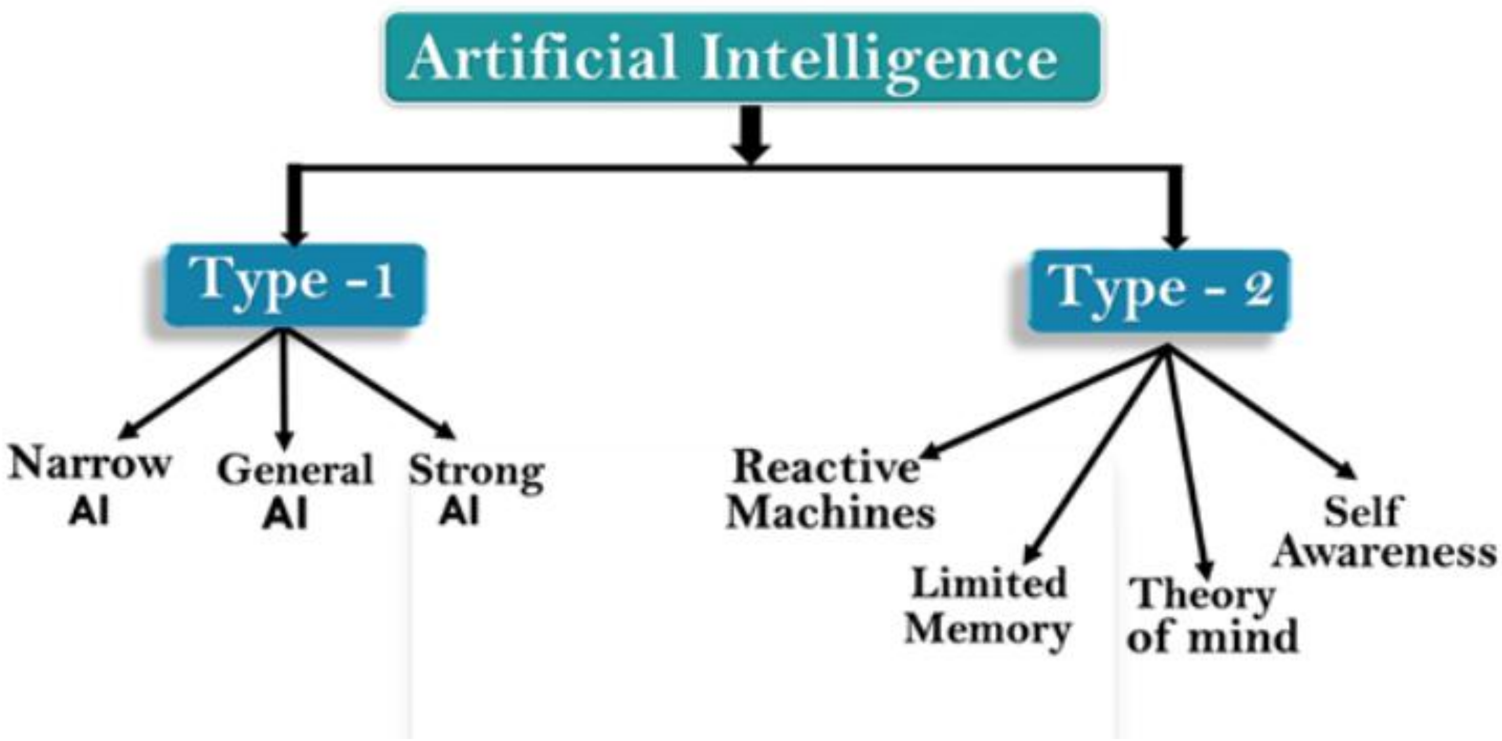
Disadvantages of Artificial Intelligence

- **High Cost:** The hardware and software requirement of AI is very costly as it requires lots of maintenance to meet current world requirements.
- **Can't think out of the box:** Even we are making smarter machines with AI, but still they cannot work out of the box, as the robot will only do that work for which they are trained, or programmed.
- **No feelings and emotions:** AI machines can be an outstanding performer, but still it does not have the feeling so it cannot make any kind of emotional attachment with human, and may sometime be harmful for users if the proper care is not taken.
- **Increase dependency on machines:** With the increment of technology, people are getting more dependent on devices and hence they are losing their mental capabilities.
- **No Original Creativity:** As humans are so creative and can imagine some new ideas but still AI machines cannot beat this power of human intelligence and cannot be creative and imaginative.

Application of AI



TYPES OF AI



Artificial Intelligence
Type -1
Type - 2
Narrow AI
General AI
Strong AI
Reactive Machines
Limited Memory
Theory of mind
Self Awareness

Types of AI

- These three types are:
- Artificial Narrow Intelligence
- Artificial General Intelligence
- Artificial Super Intelligence

Weak AI or Narrow AI:

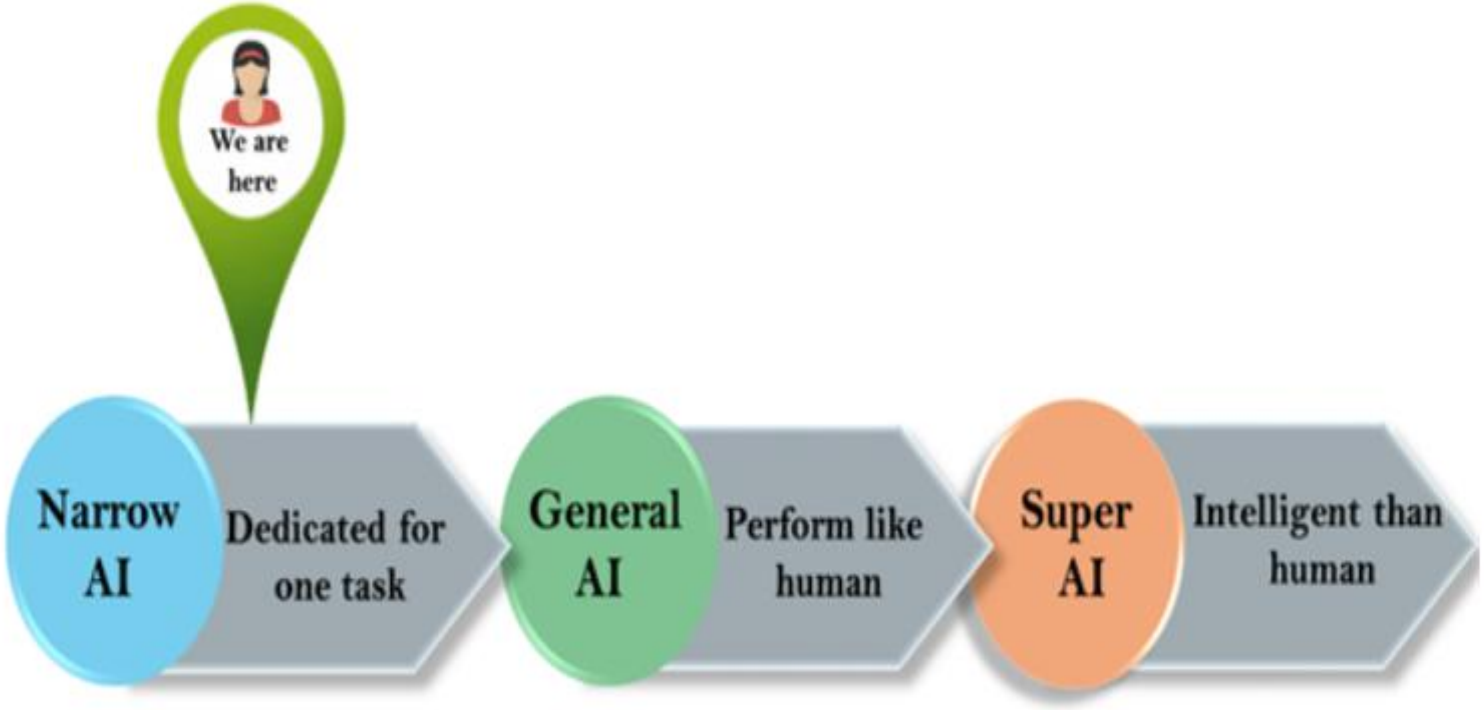
- Narrow AI is a type of AI which is able to perform a dedicated task with intelligence.The most common and currently available AI is Narrow AI in the world of Artificial Intelligence.
- Narrow AI cannot perform beyond its field or limitations, as it is only trained for one specific task. Hence it is also termed as weak AI. Narrow AI can fail in unpredictable ways if it goes beyond its limits.
- Apple Siriis a good example of Narrow AI, but it operates with a limited pre-defined range of functions.

General AI:

- General AI is a type of intelligence which could perform any intellectual task with efficiency like a human.
- The idea behind the general AI to make such a system which could be smarter and think like a human by its own.
- Currently, there is no such system exist which could come under general AI and can perform any task as perfect as a human.

Super AI:

- Super AI is a level of Intelligence of Systems at which machines could surpass human intelligence, and can perform any task better than human with cognitive properties. It is an outcome of general AI.
- Some key characteristics of strong AI include capability include the ability to think, to reason,solve the puzzle, make judgments, plan, learn, and communicate by its own.
- Super AI is still a hypothetical concept of Artificial Intelligence. Development of such systems in real is still world changing task.


We are here
Narrow AI
Dedicated for one task
General AI
Perform like human
Super AI
Intelligent than human

- Reactive Machines
- Purely reactive machines are the most basic types of Artificial Intelligence.
- Such AI systems do not store memories or past experiences for future actions.
- These machines only focus on current scenarios and react on it as per possible best action.

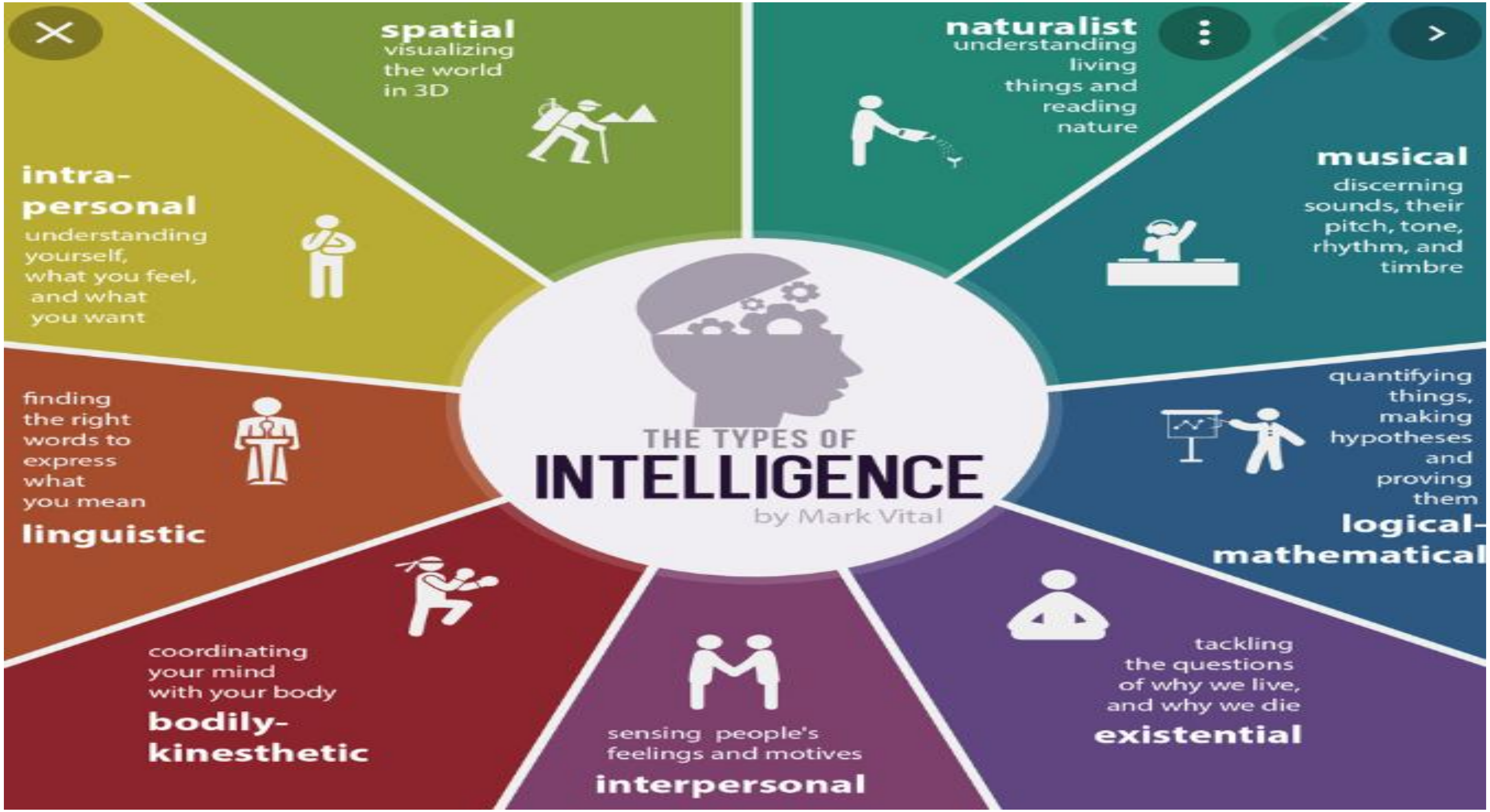
LIMITED MEMORY

- Limited memory machines can store past experiences or some data for a short period of time.
- These machines can use stored data for a limited time period only.
- Self-driving cars are one of the best examples of Limited Memory systems. These cars can store recent speed of nearby cars, the distance of other cars, speed limit, and other information to navigate the road.

THEORY OF MIND

- Theory of Mind AI should understand the human emotions, people, beliefs, and be able to interact socially like humans.
- This type of AI machines are still not developed, but researchers are making lots of efforts and improvement for developing such AI machines.

- Self-Awareness
- Self-awareness AI is the future of Artificial Intelligence. These machines will be super intelligent, and will have their own consciousness, sentiments, and self-awareness.
- These machines will be smarter than human mind.
- Self-Awareness AI does not exist in reality still and it is a hypothetical concept.


spatial
visualizing the world in 3D
naturalist
understanding living things and reading nature
musical
discerning sounds, their pitch, tone, rhythm, and timbre
quantifying things, making hypotheses and proving them
logical-mathematical
tackling the questions of why we live, and why we die
existential
sensing people's feelings and motives
interpersonal
coordinating your mind with your body
bodily-kinesthetic
finding the right words to express what you mean
linguistic
intra-personal
understanding yourself, what you feel, and what you want
THE TYPES OF
INTELLIGENCE
by Mark Vital

SPACIAL INTELLIGENCE

Strengths
Visual and spacial judgment

Characteristics
-Draws for fun
-Good at puzzles
-Recognizes patterns
-Interprets visuals well

BODILY-KINESTHETIC INTELLIGENCE

Strengths
Physical movement, motor control

Characteristics
-Skilled at sports
-Excellent physical coordination
-Remembers by doing, instead of hearing or seeing

MUSICAL INTELLIGENCE

Strengths
Rhythm and music

Characteristics
-Appreciation for music
-Thinks in sounds and patterns
-Rich understanding of musical structure, notes

LINGUISTIC INTELLIGENCE

Strengths
Words, language, writing

Characteristics
-Enjoys writing, reading
-Good at public speaking
-Very persuasive
-Can explain things well

LOGICAL-MATHEMATICAL SKILLS

Strengths
Analyzing problems, mathematical operations

Characteristics
-Fast problem-solver
-Understands complex computations
-Likes thinking about abstract ideas

INTERPERSONAL INTELLIGENCE

Strengths
Understanding and relating to others

Characteristics
-Strong emotional intelligence skills
-Creates healthy relationships
-Good at solving conflicts

INTRAPERSONAL INTELLIGENCE

Strengths
Introspection and self-reflection

Characteristics
-Understands one's own strengths, weaknesses
-Highly self-aware
-Sensitive to one's own feelings

NATURALISTIC INTELLIGENCE

Strengths
Seeing patterns and relationships to nature

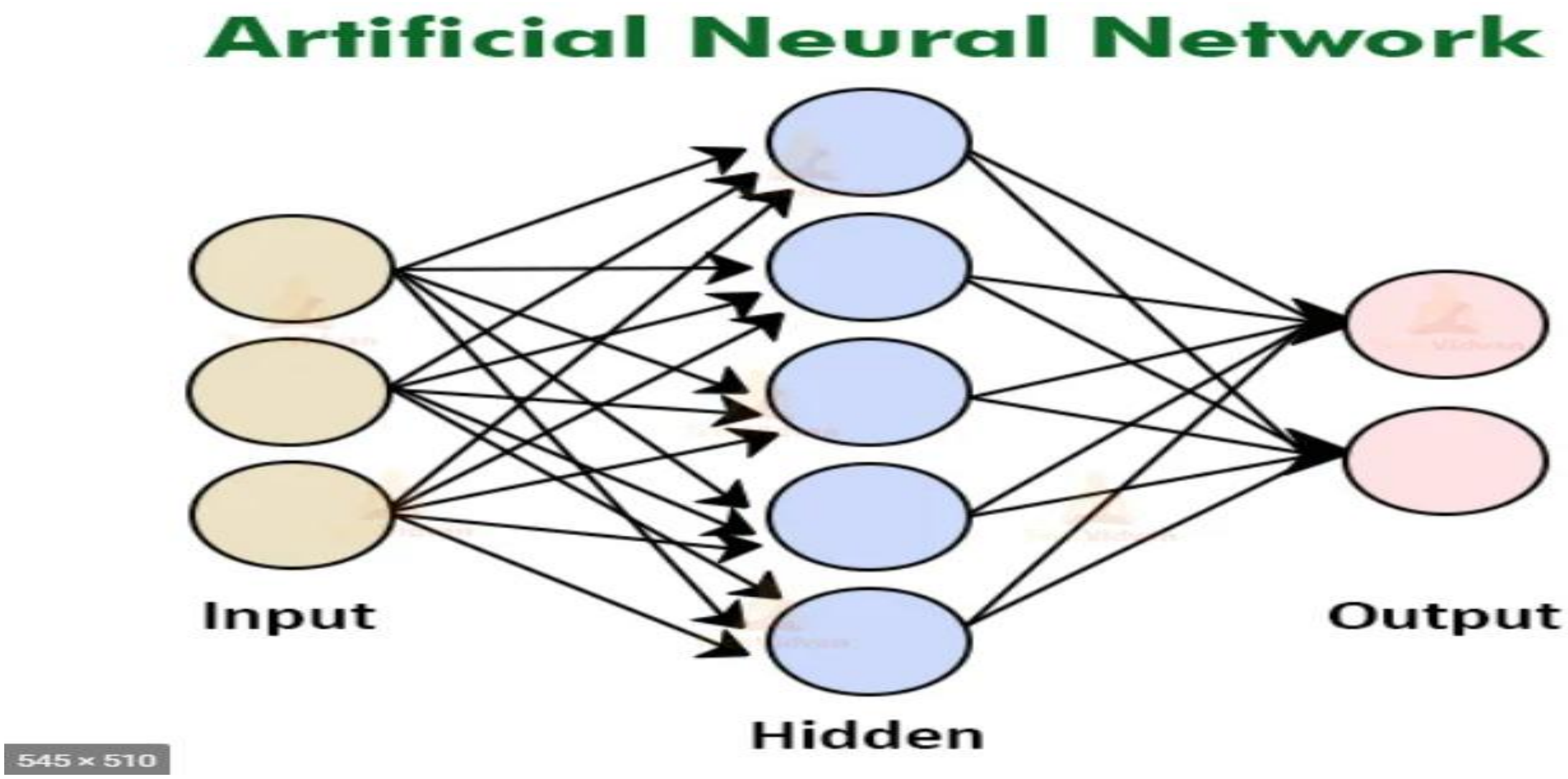
Characteristics
-Interested in areas like botany, biology, zoology
-Appreciation for nature
-Enjoys activities like camping, gardening, hiking

NUERAL NETWORKS

What is Nueral Networks

- A neural network is **a method in artificial intelligence that teaches computers to process data in a way that is inspired by the human brain**. It is a type of machine learning process, called deep learning, that uses interconnected nodes or neurons in a layered structure that resembles the human brain.

Architecture of NN



Types of NN

- ANN
- CNN
- RNN

ANN

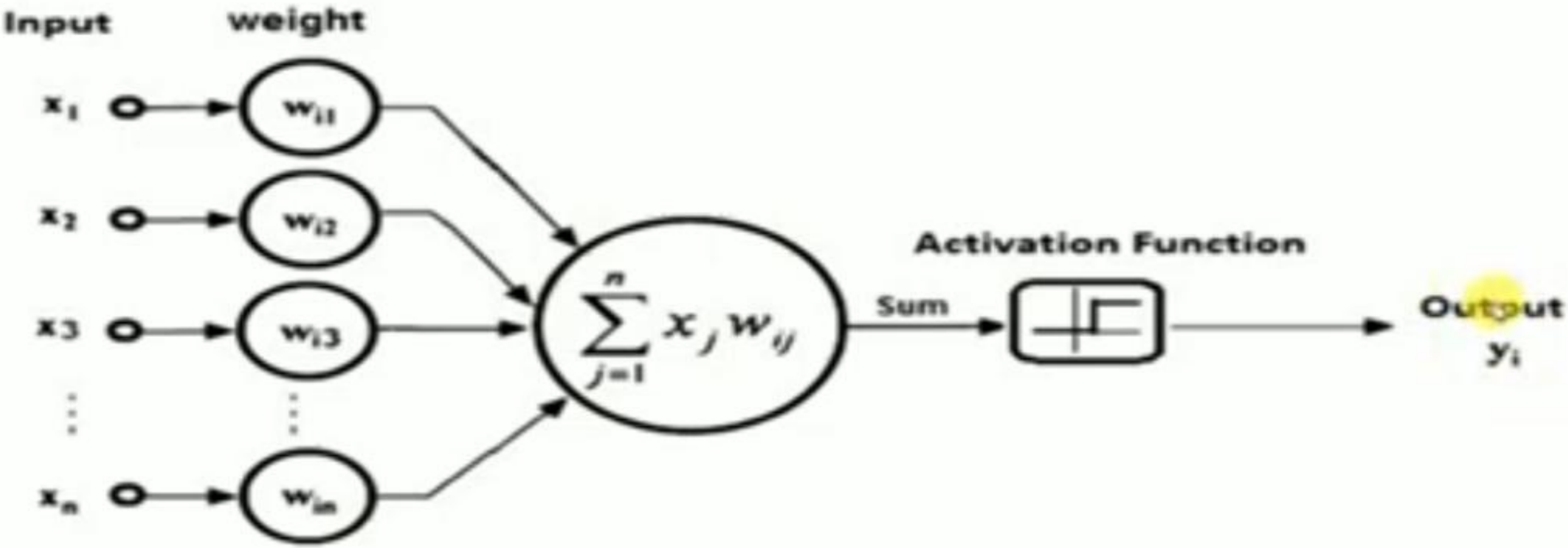
Artificial Neural Networks

- Artificial neural networks (ANNs) are inspired by the information processing model of the human mind/brain.
- The human brain consists of billions of neurons that link with one another in an intricate pattern.
- Every neuron receives information from many other neurons, processes it, gets excited or not, and passes its state information to other neurons.

Business Applications of ANN

1. They are used in stock price prediction where the rules of the game are extremely complicated, and a lot of data needs to be processed very quickly.
2. They are used for character recognition, as in recognizing handwritten text, or damaged or mangled text.
3. They are used in recognizing finger prints. These are complicated patterns and are unique for each person. Layers of neurons can progressively clarify the pattern.
4. They are also used in traditional classification problems, such as approving a loan application.

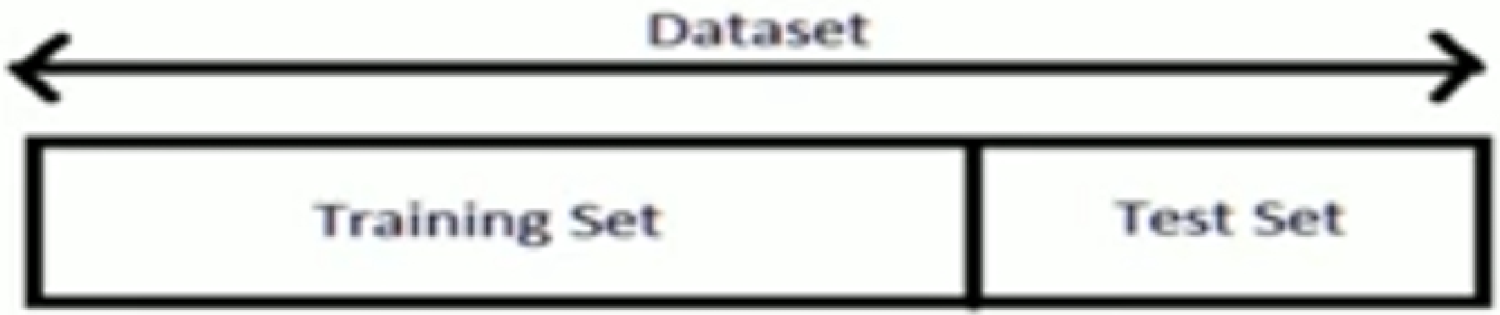
Artificial neural network (ANN)



Advantages of Using ANNs

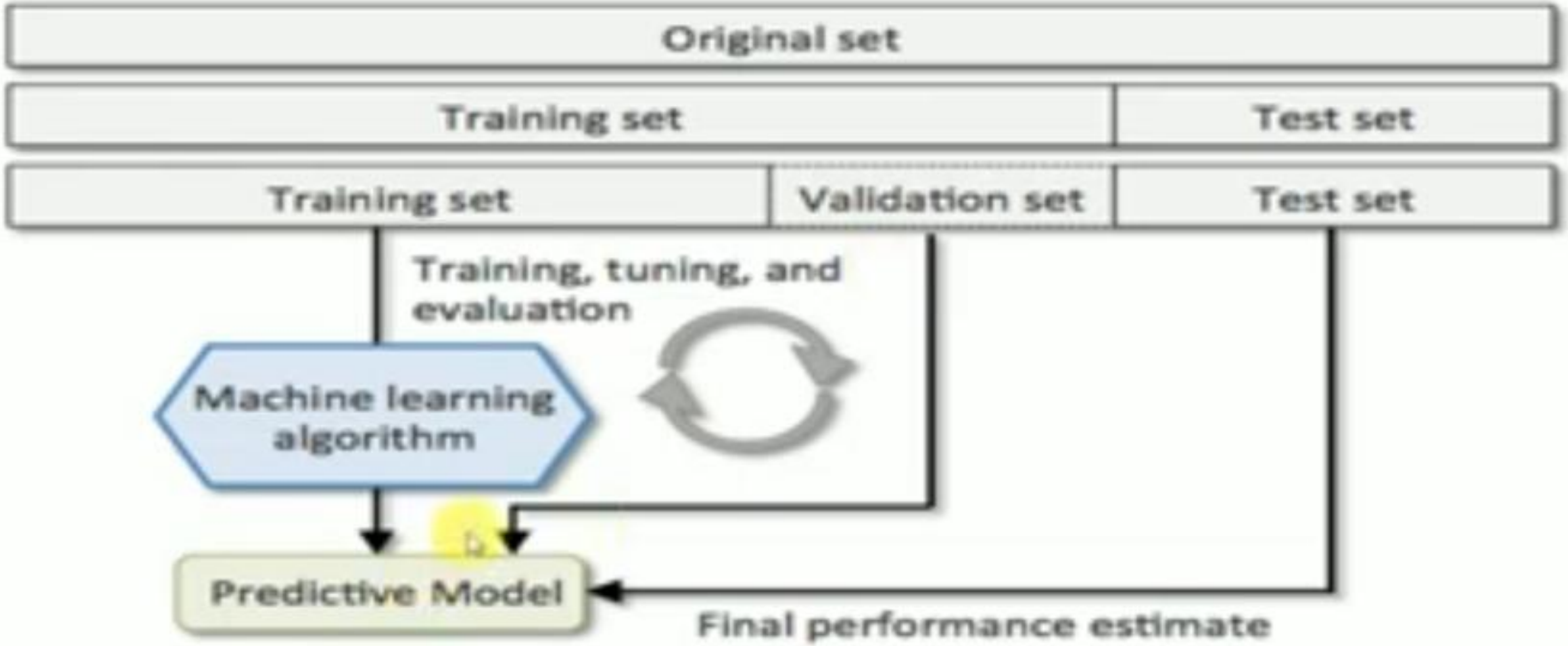
1. ANNs impose very little restrictions on their use.
2. There is no need to program ANN neural networks, as they learn from examples. They get better with use, without much programing effort.
3. ANN can handle a variety of problem types, including classification, clustering, associations, and so on.
4. ANNs are tolerant of data quality issues, and they do not restrict the data to follow strict normality and/or independence assumptions.
5. ANN can handle both numerical and categorical variables.
6. ANNs can be much faster than other techniques.
7. Most importantly, ANN usually provide better results (prediction and/or clustering) compared to statistical counterparts, once they have been trained enough.

Train Test Distribution



Train Test Validation

Train Test Validation

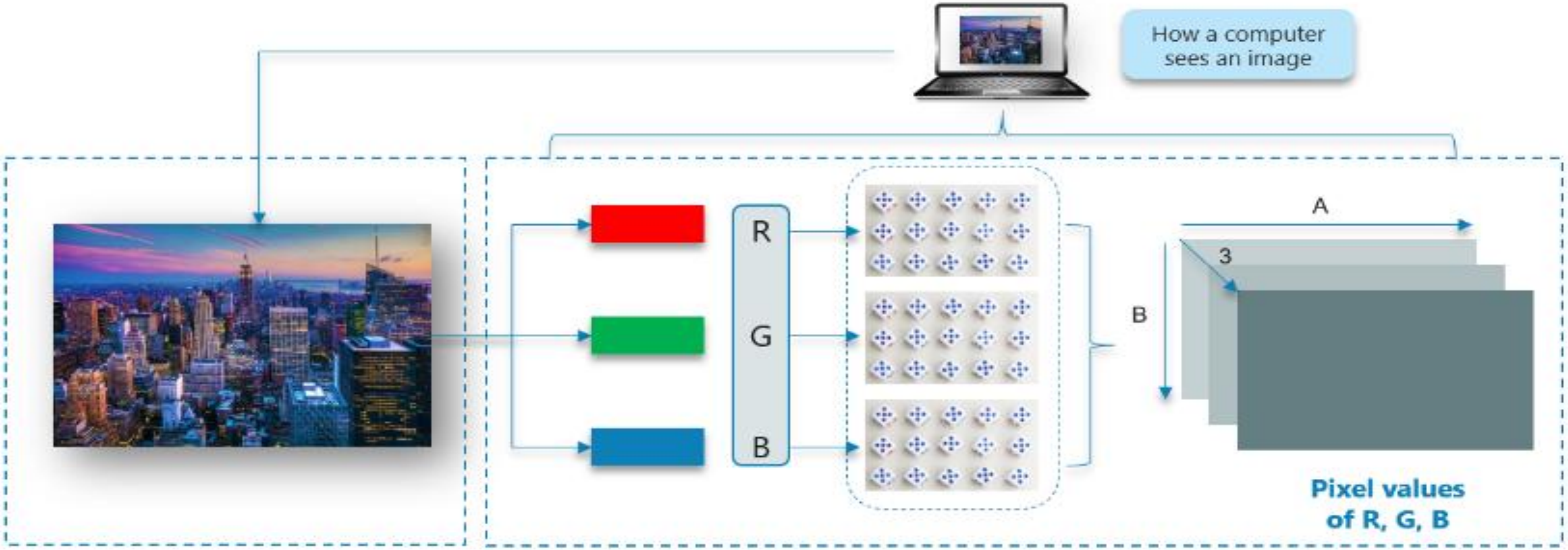


CNN

Within Deep Learning, a Convolutional Neural Network or CNN is **a type of artificial neural network, which is widely used for image/object recognition and classification**. Deep Learning thus recognizes objects in an image by using a CNN.

How Does A Computer Read an Image?
Consider this image of the **New York skyline**, upon first glance you will see a lot of **buildings** and **colors.** So how does the computer **process** this image?

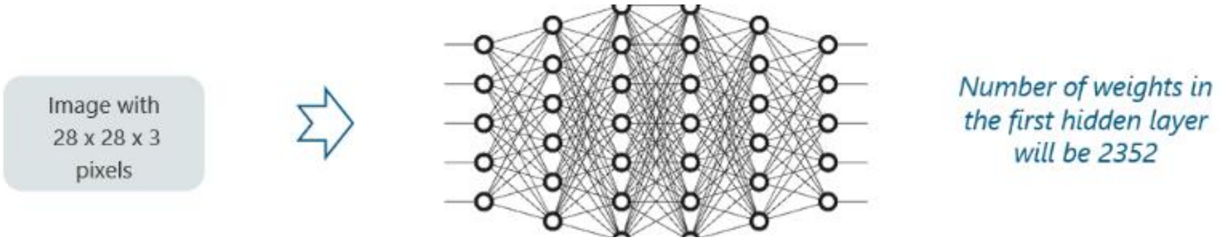
The image is **broken down** into 3 color-channels which is **Red** and **Blue.** Each of these color channels are **mapped** to the **image's pixel.**


How a computer sees an image
R
G
B
A
3
B
Pixel values of R, G, B

Why Not Fully Connected Networks?

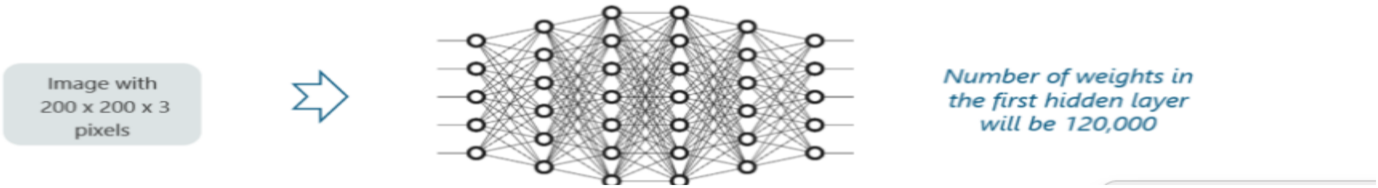
We **cannot** make use of fully connected networks when it comes to **Convolutional Neural Networks,** here's why!

Consider the following image:



Here, we have **considered** an **input** of images with the size **28x28x3** pixels. If we **input** this to our Convolutional Neural Network, we will have about **2352 weights** in the **first** hidden layer itself.

But this case **isn't practical**. Now, take a look at this:



Any **generic** input **image** will **atleast** have **200x200x3 pixels** in size. The size of the first hidden layer becomes a **whooping 120,000**. If this is just the **first** hidden layer, imagine the **number of neurons** needed to process an **entire** complex **image-set.**

This leads to **over-fitting** and isn't practical. **Hence, we cannot make use of fully connected networks.**

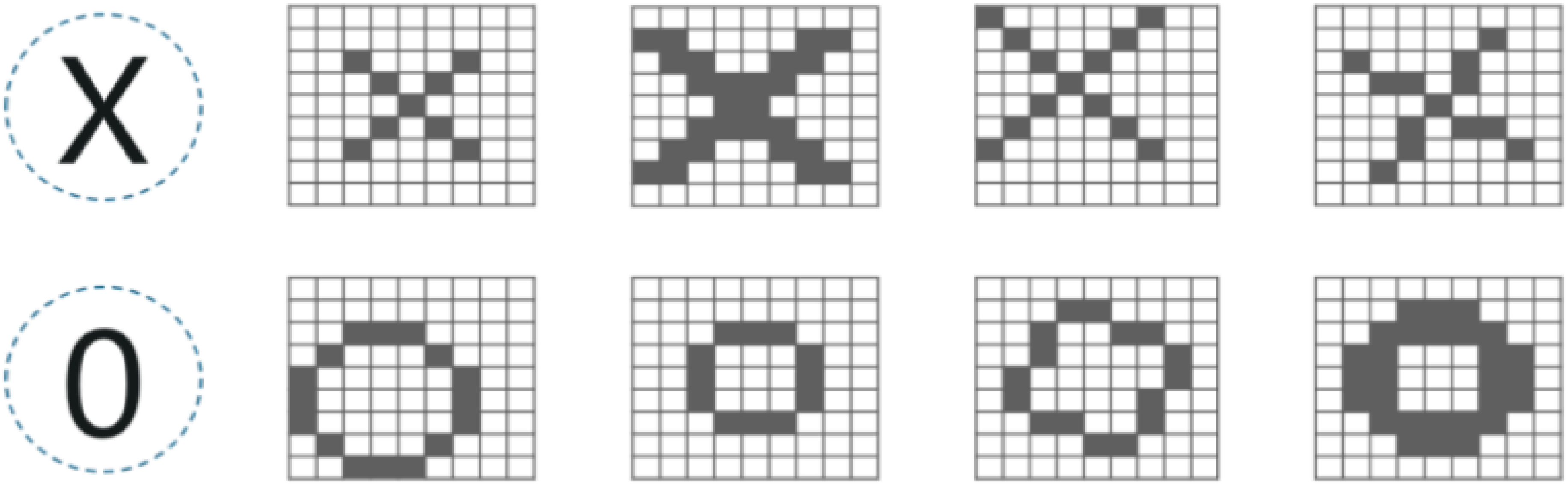
How Do Convolutional Neural Networks Work?

There are **four** layered **concepts** we should understand in Convolutional Neural Networks:

1. Convolution,
2. ReLu,
3. Pooling and
4. Full Connectedness (Fully Connected Layer).

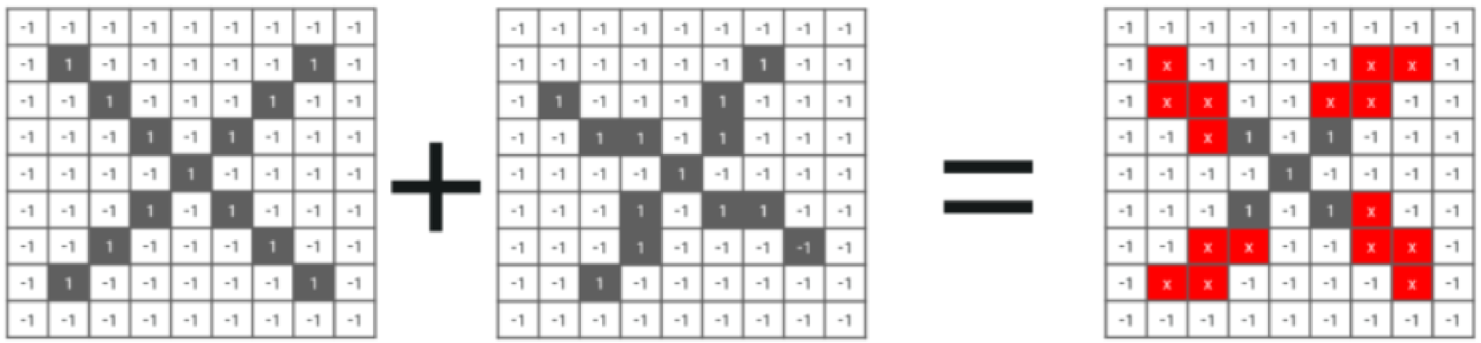
Example of CNN:

Consider the image below:



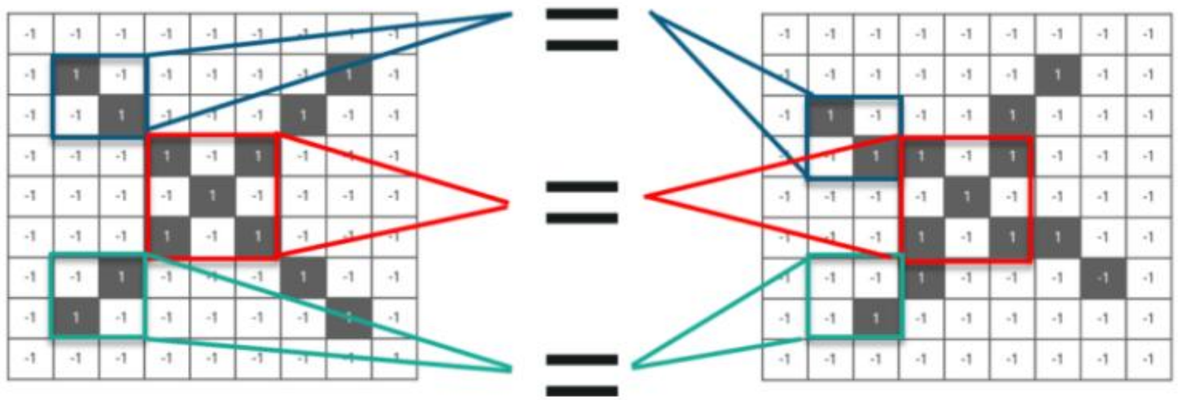
-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

So, the **computer understands** every pixel. In this case, the **white** pixels are said to be **-1** while the **black** ones are **1.** This is just the way we've implemented to **differentiate the pixels** in a basic binary classification.



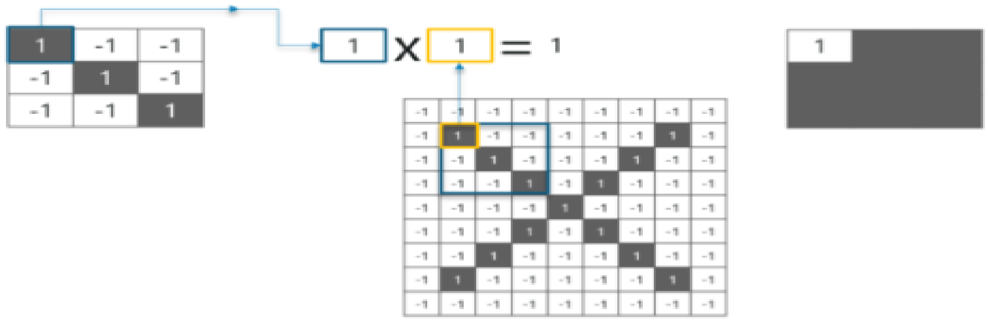
Now if we would just **normally search** and **compare** the **values** between a normal image and another **'x' rendition,** we would get a **lot** of **missing pixels.**

So, how do we fix this?

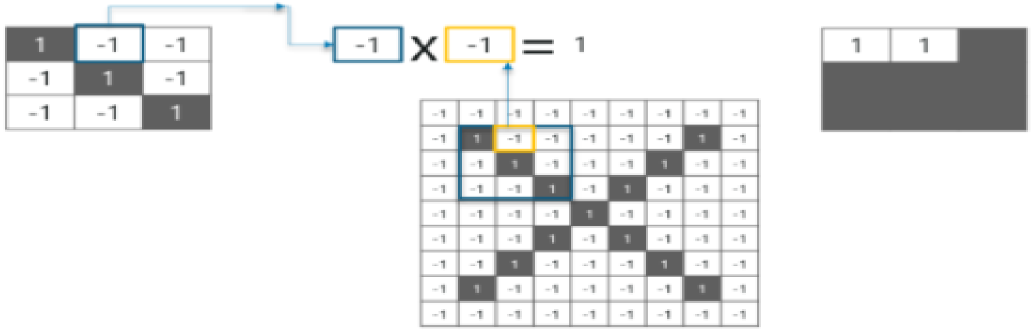


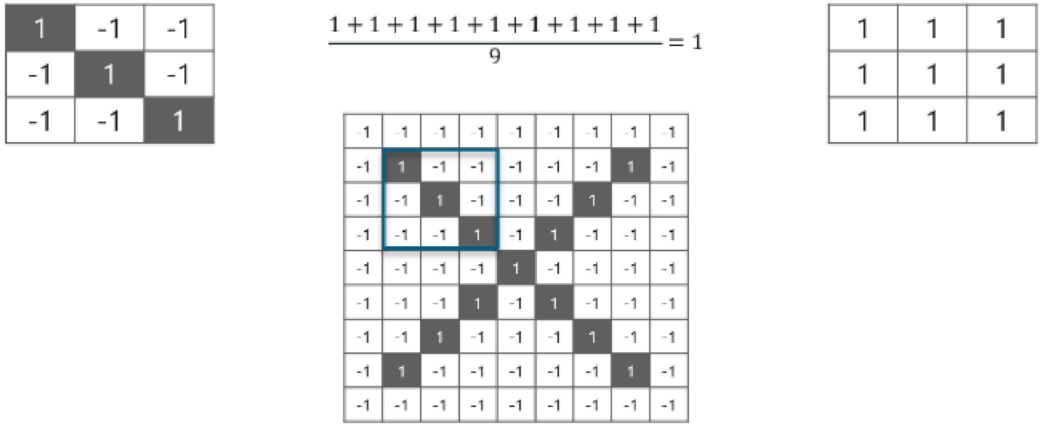
Convolution: it has 4 steps

- **Line up** the feature and the image
- **Multiply** each **image** pixel by corresponding **feature** pixel
- **Add** the values and find the **sum**
- **Divide** the sum by the **total** number of pixels in the **feature**

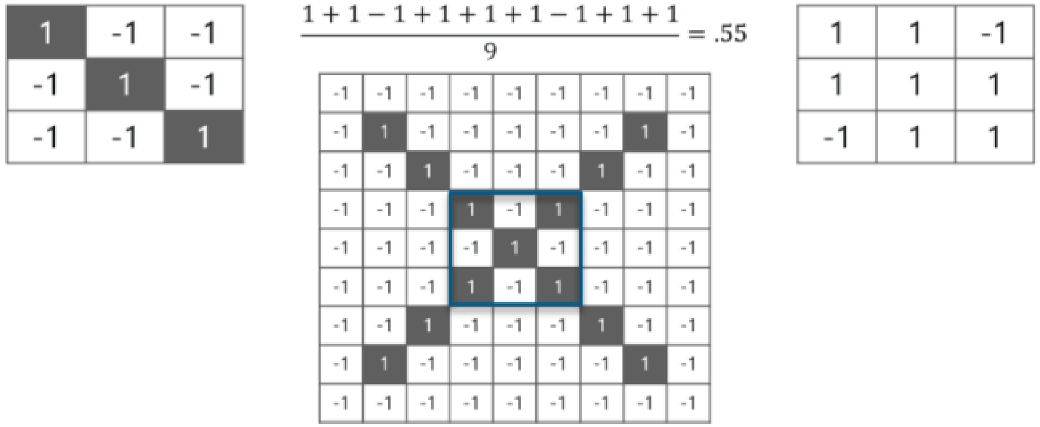


Consider the above image – As you can see, we are **done** with the first **2 steps**. We considered a **feature image** and **one pixel** from it. We **multiplied** this with the **existing image** and the product is stored in another **buffer feature image**.





Now, we can **move** this **filter** around and do the **same** at **any pixel** in the image. For **better clarity,** let's consider **another example:**



Similarly, we move the feature to every other position in the image and see how the feature matches that area. So after doing this, we will get the output as:

0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.0	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.0	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

Here we considered just one filter. Similarly, we will perform the same convolution with every other filter to get the convolution of that filter.

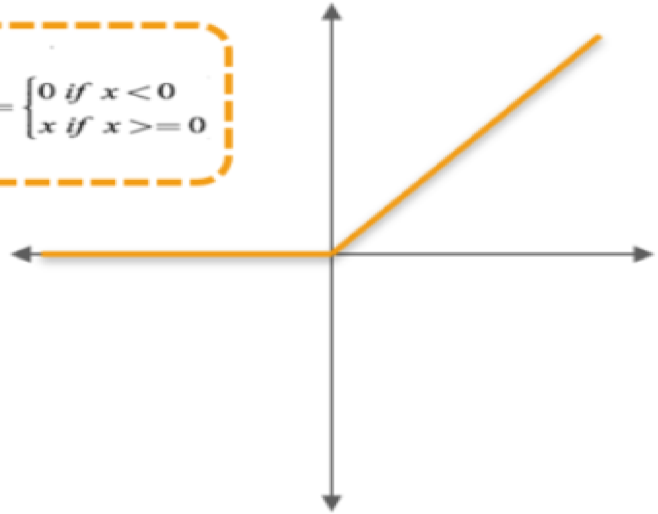
ReLU Layer

ReLU is an activation function. But, what is an activation function?

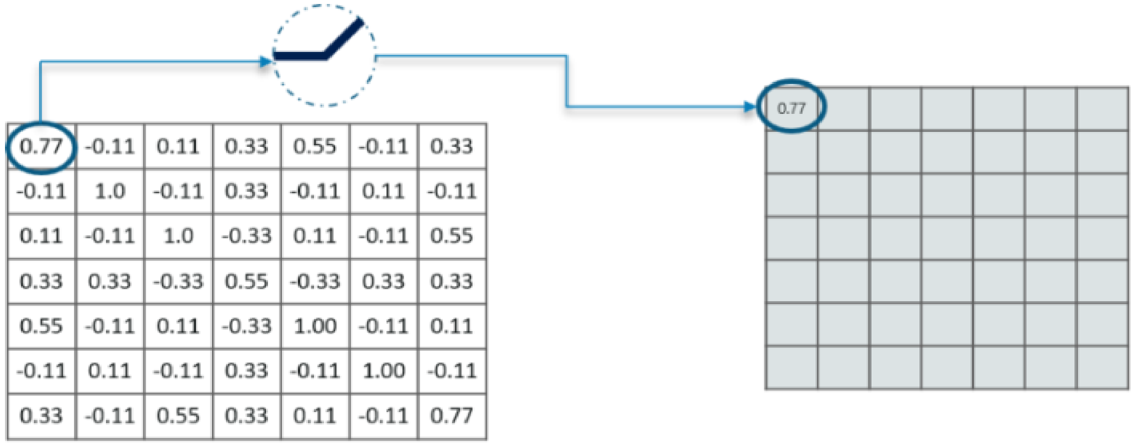
Rectified Linear Unit (ReLU) transform function only activates a node if the input is above a certain quantity, while the input is below zero, the output is zero, but when the input rises above a certain threshold, it has a linear relationship with the dependent variable.

Consider the below example:

$$f(x)=\begin{cases}0 \text{ if } x<0\\ x \text{ if } x>=0\end{cases}$$



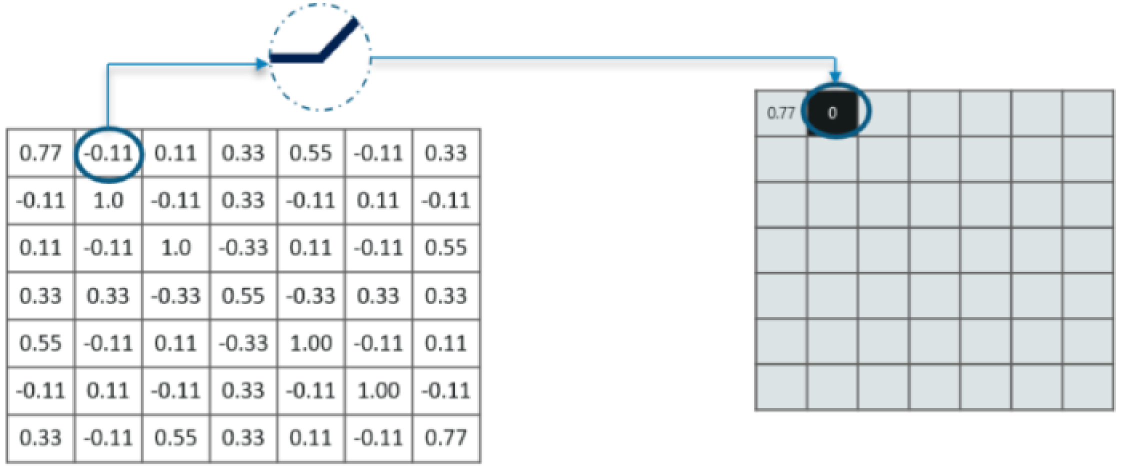
Why do we require ReLU here?



0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.0	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.0	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

The main aim is to remove all the negative values from the convolution. All the positive values remain the same but all the negative values get changed to zero as shown below:

The main aim is to remove all the negative values from the convolution. All the positive values remain the same but all the negative values get changed to zero as shown below:



0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.0	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.0	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

Now, similarly we do the same process to all the other feature images as well:

0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.0	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.0	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

0.77	0	0.11	0.33	0.55	0	0.33
0	1.00	0	0.33	0	0.11	0
0.11	0	1.00	0	0.11	0	0.55
0.33	0.33	0	0.55	0	0.33	0.33
0.55	0	0.11	0	1.00	0	0.11
0	0.11	0	0.33	0	1.00	0
0.33	0	0.55	0.33	0.11	0	1.77

0.33	-0.55	0.11	-0.11	0.11	-0.55	0.33
-0.55	0.55	-0.55	0.33	-0.55	0.55	-0.55
0.11	-0.55	0.55	-0.11	0.55	-0.55	0.11
-0.11	0.33	-0.77	1.00	-0.77	0.33	-0.11
0.11	-0.55	0.55	-0.77	0.55	-0.55	0.11
-0.55	0.55	-0.55	0.33	-0.55	0.55	-0.55
0.33	-0.55	0.11	-0.11	0.11	-0.55	0.33

0.33	0	0.11	0	0.11	0	0.33
0	0.55	0	0.33	0	0.55	0
0.11	0	0.55	0	0.55	0	0.11
0	0.33	0	1.00	0	0.33	0
0.11	0	0.55	0	0.55	0	0.11
0	0.55	0	0.33	0	0.55	0
0.33	0	0.11	0	0.11	0	0.33

0.33	-0.11	0.55	0.33	0.11	-0.11	0.77
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.77	-0.11	0.11	0.33	0.55	-0.11	0.33

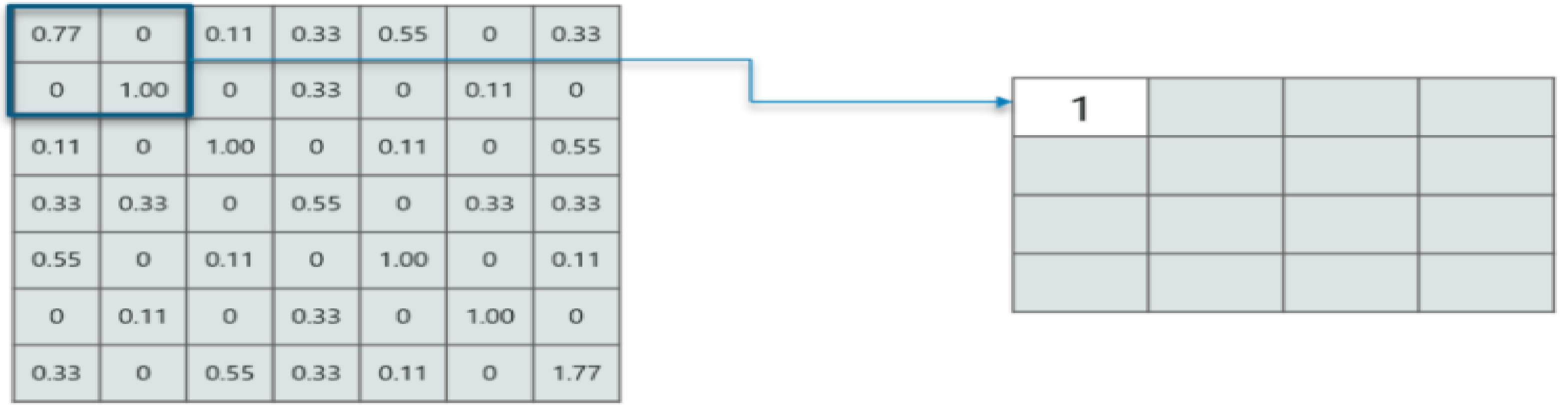
0.33	0	0.55	0.33	0.11	0	0.77
0	0.11	0	0.33	0	1.00	0
0.55	0	0.11	0	1.00	0	0.11
0.33	0.33	0	0.55	0	0.33	0.33
0.11	0	1.00	0	0.11	0	0.55
0	1.00	0	0.33	0	0.11	0
0.77	0	0.11	0.33	0.55	0	0.33

Pooling Layer

In this layer we **shrink** the **image** stack into a **smaller size.** Pooling is done **after passing** through the **activation** layer. We do this by implementing the following 4 steps:

- Pick a **window size** (usually 2 or 3)
- Pick a **stride** (usually 2)
- **Walk** your window **across** your **filtered** images
- From each **window,** take the **maximum** value

Let us understand this with an example. Consider performing pooling with a window size of 2 and stride being 2 as well.



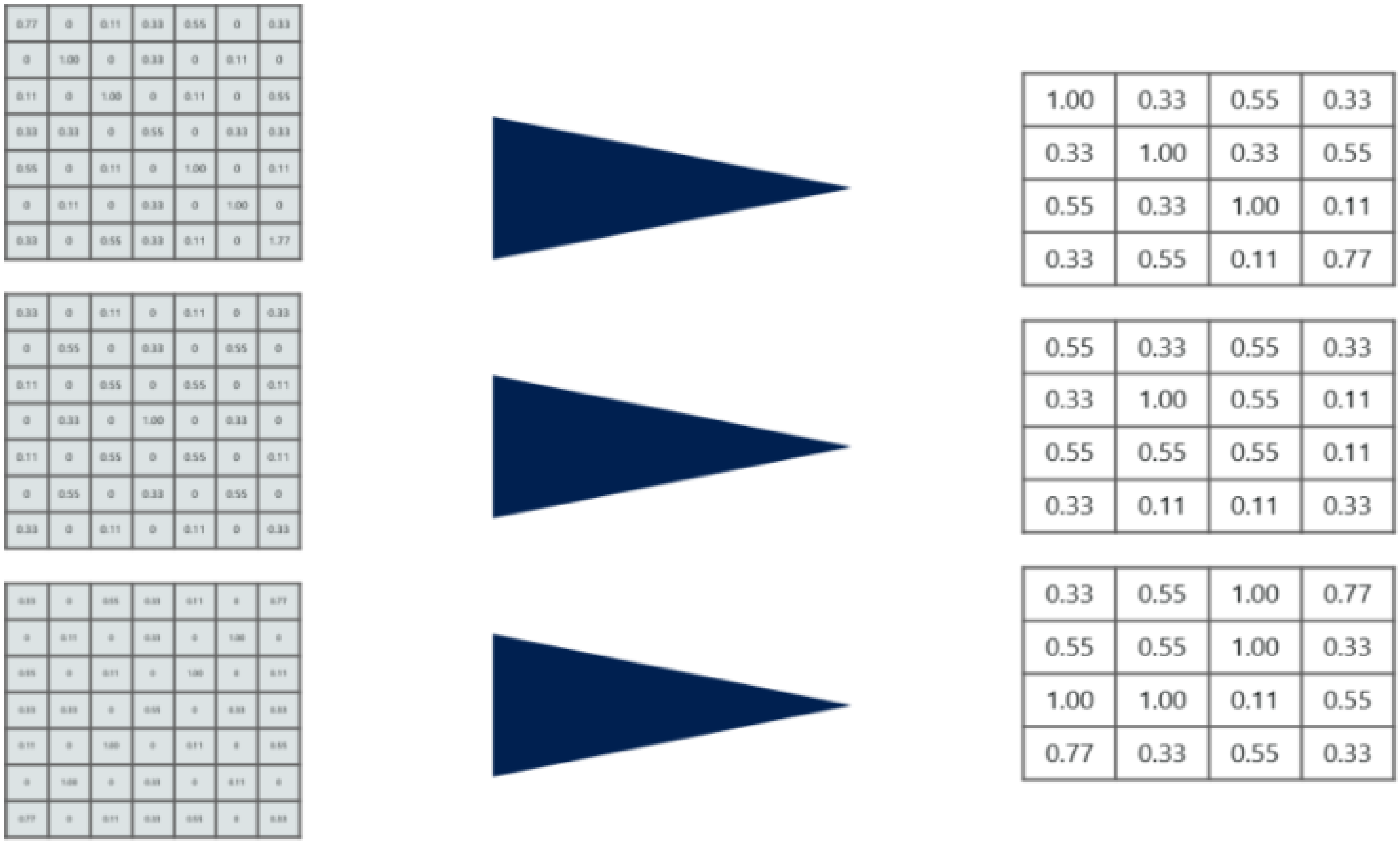
So in this case, we took **window size** to be **2** and we got **4 values** to choose from. From those 4 values, the **maximum value** there is 1 so we pick 1. Also, note that we **started out** with a **7×7** matrix but now the same matrix after **pooling** came down to **4×4.**

But we need to **move** the **window across** the **entire** image. The procedure is exactly as same as above and we need to repeat that for the entire image.

0.77	0	0.11	0.33	0.55	0	0.33
0	1.00	0	0.33	0	0.11	0
0.11	0	1.00	0	0.11	0	0.55
0.33	0.33	0	0.55	0	0.33	0.33
0.55	0	0.11	0	1.00	0	0.11
0	0.11	0	0.33	0	1.00	0
0.33	0	0.55	0.33	0.11	0	1.77

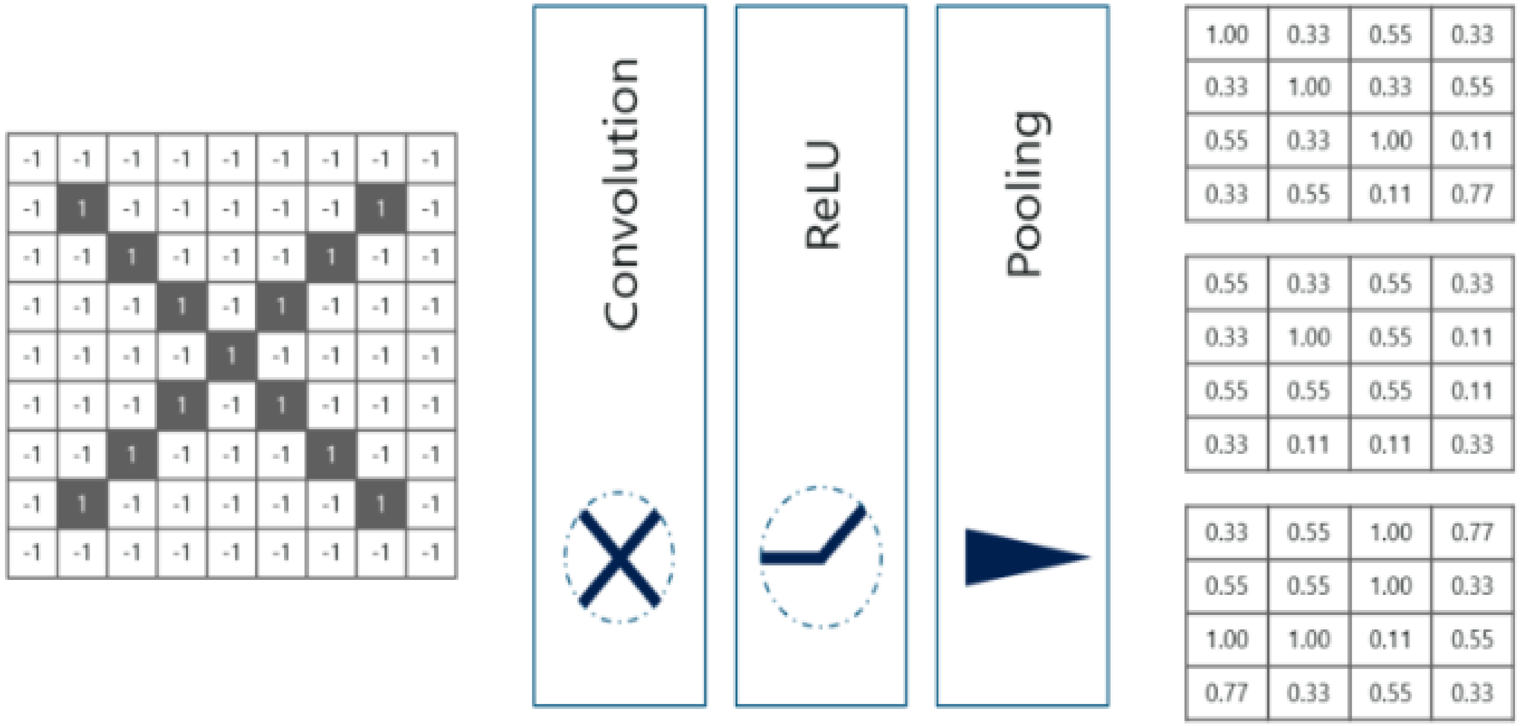
1.00	0.33	0.55	0.33
0.33	1.00	0.33	0.55
0.55	0.33	1.00	0.11
0.33	0.55	0.11	0.77

Do note that this is for **one filter.** We need to do it for 2 other filters as well. This is done and we arrive at the following result:



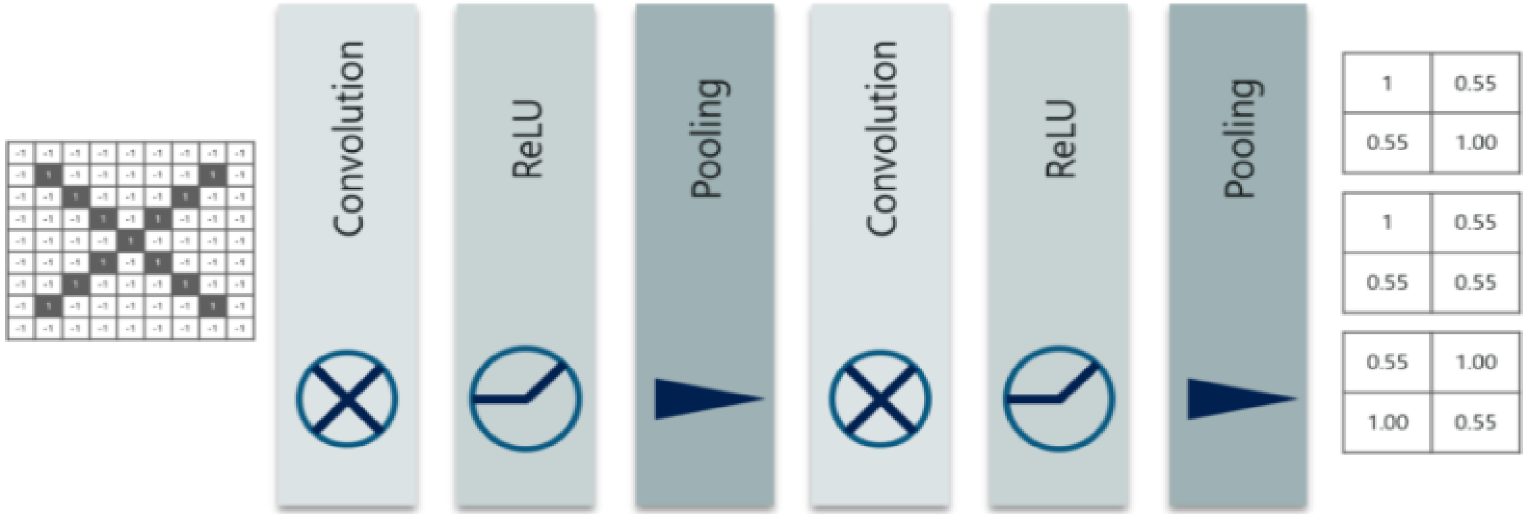
Stacking Up The Layers

So to get the **time-frame** in one picture we're here with a **4×4** matrix from a **7×7** matrix after passing the input through 3 layers – **Convolution, ReLU** and **Pooling** as shown below:



But can we **further reduce** the image from **4×4** to **something lesser?**

Yes, we can! We need to perform the 3 operations in an iteration after the first pass. So after the second pass we arrive at a 2×2 matrix as shown below:



The last layers in the network are **fully connected,** meaning that neurons of preceding layers are **connected** to **every neuron** in **subsequent** layers.

This **mimics high level reasoning** where all possible **pathways** from the **input** to **output** are considered.

Also, fully connected layer is the final layer where the classification actually happens. Here we take our filtered and shrinked images and put them into one single list as shown below:

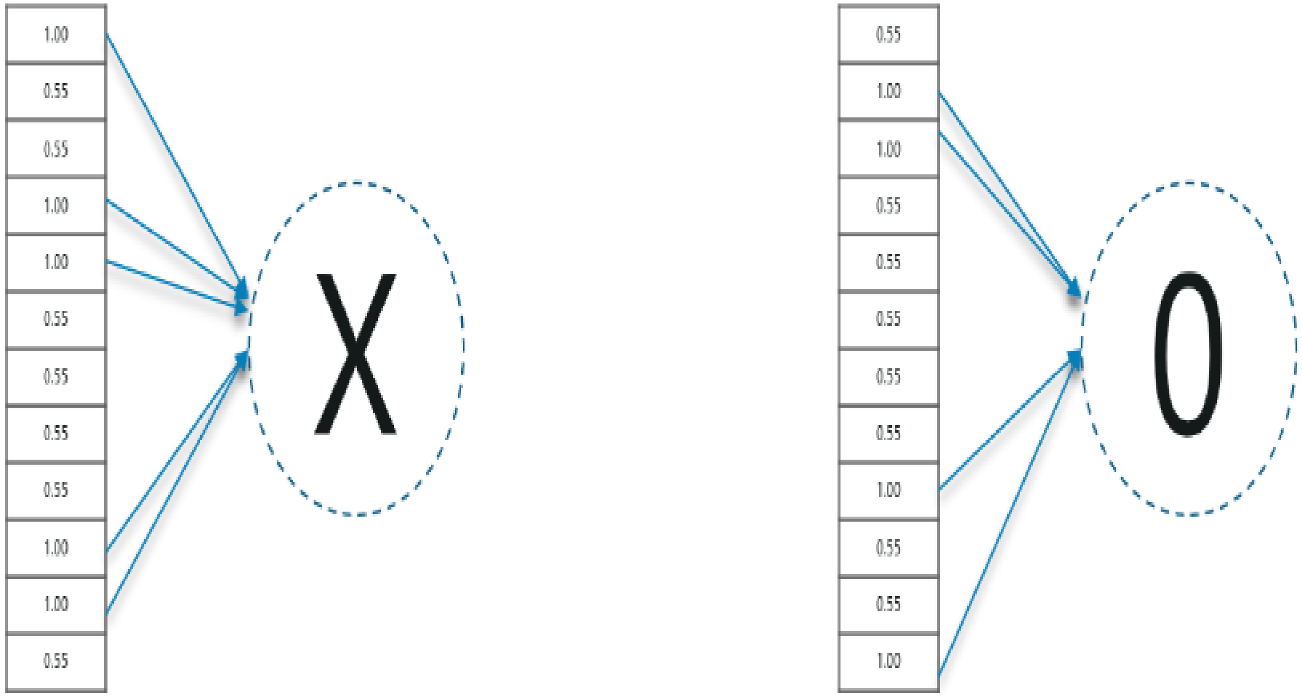
1	0.55
0.55	1.00

1	0.55
0.55	0.55

0.55	1.00
1.00	0.55

1.00
0.55
0.55
1.00
1.00
0.55
0.55
0.55
0.55
1.00
1.00
0.55

So **next,** when we feed in, **'X'** and **'O'** there will be **some element** in the vector that will be **high.** Consider the image below, as you can see for 'X' there are **different elements** that are **high** and **similarly,** for **'O'** we have **different elements** that are **high:**

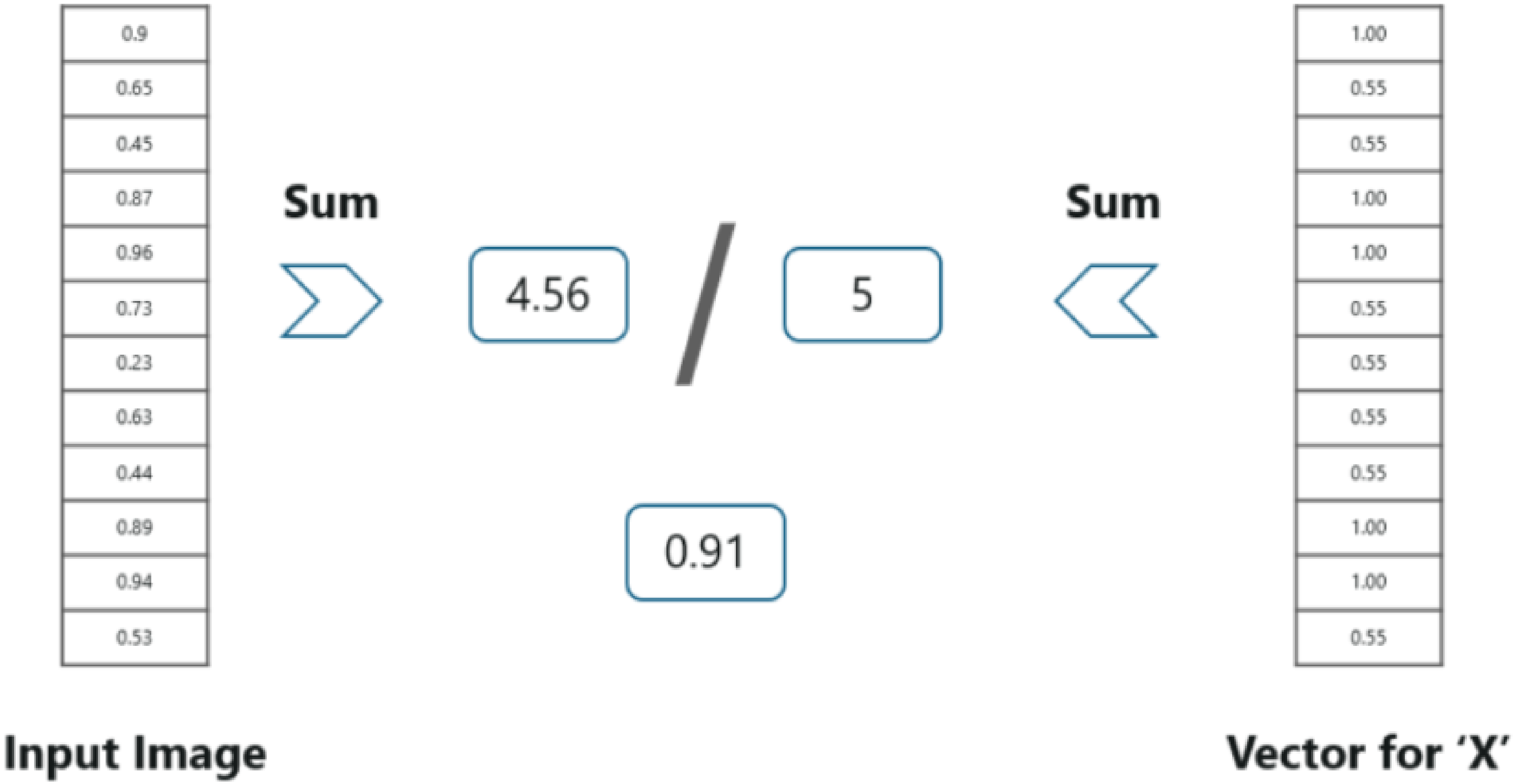


Prediction Of Image Using Convolutional Neural Networks – Fully Connected Layer

At this point in time, **we’re done training** the network and we can begin to predict and **check** the **working** of the **classifier**. Let’s check out a simple example:

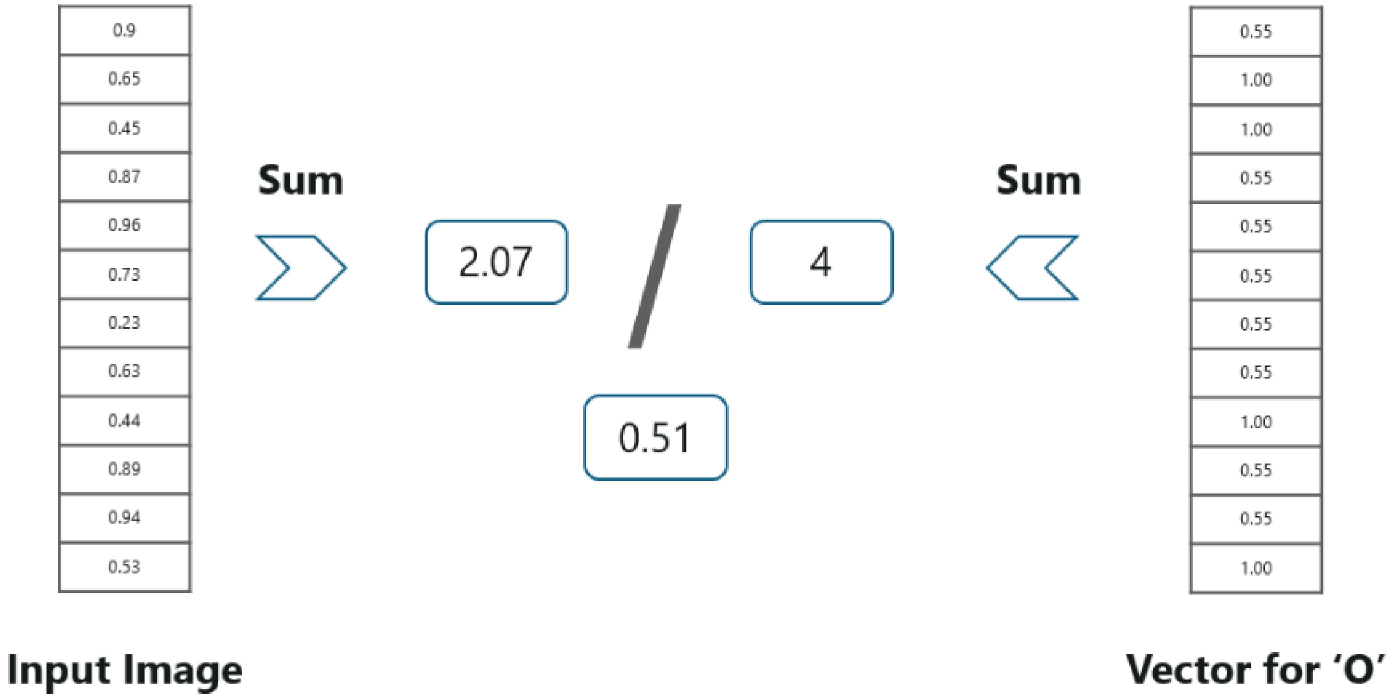
0.9
0.65
0.45
0.87
0.96
0.73
0.23
0.63
0.44
0.89
0.94
0.53

We **make predictions** based on the **output** data by comparing the **obtained values** with list of 'x'and 'o'!



Well, it is **really easy.** We just **added** the values we which found out as high (1st, 4th, 5th, 10th and 11th) from the **vector table** of **X** and we got the sum to be **5.** We did the **exact same thing** with the **input image** and got a value of **4.56**.

When we **divide** the **value** we have a **probability match** to be **0.91!** Let’s do the **same** with the **vector table** of **‘o’** now:



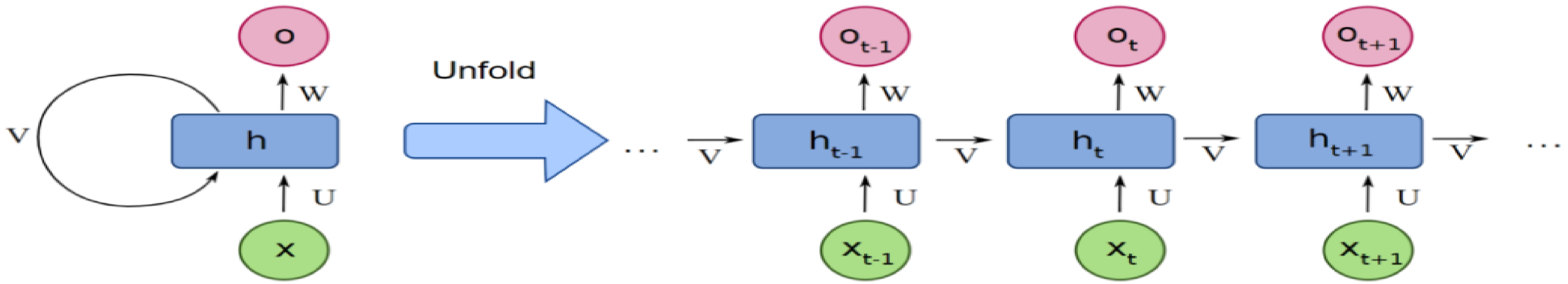
We have the **output** as **0.51** with this table. Well, probability being **0.51** is less than **0.91**, isn’t it?

So we can conclude that the **resulting input image** is an **‘x’!**

RNN

Introduction on Recurrent Neural Networks

A Deep Learning approach for modelling sequential data is **Recurrent Neural Networks (RNN)**. RNNs were the standard suggestion for working with sequential data before the advent of attention models. Specific parameters for each element of the sequence may be required by a deep feedforward model. It may also be unable to generalize to variable-length sequences.

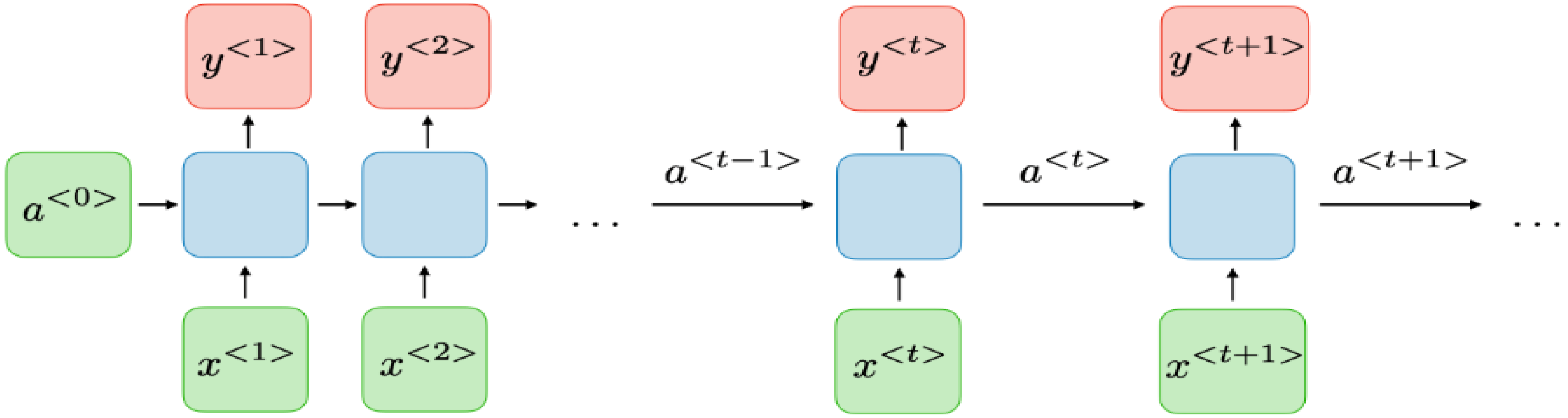


Recurrent Neural Networks use the same weights for each element of the sequence, decreasing the number of parameters and allowing the model to generalize to sequences of varying lengths. RNNs generalize to structured data other than sequential data, such as geographical or graphical data, because of its design.

Recurrent neural networks, like many other deep learning techniques, are relatively old. They were first developed in the 1980s, but we didn't appreciate their full potential until lately. The advent of long short-term memory (LSTM) in the 1990s, combined with an increase in computational power and the vast amounts of data that we now have to deal with, has really pushed RNNs to the forefront.

The Architecture of a Traditional RNN

RNNs are a type of neural network that has hidden states and allows past outputs to be used as inputs. They usually go like this:



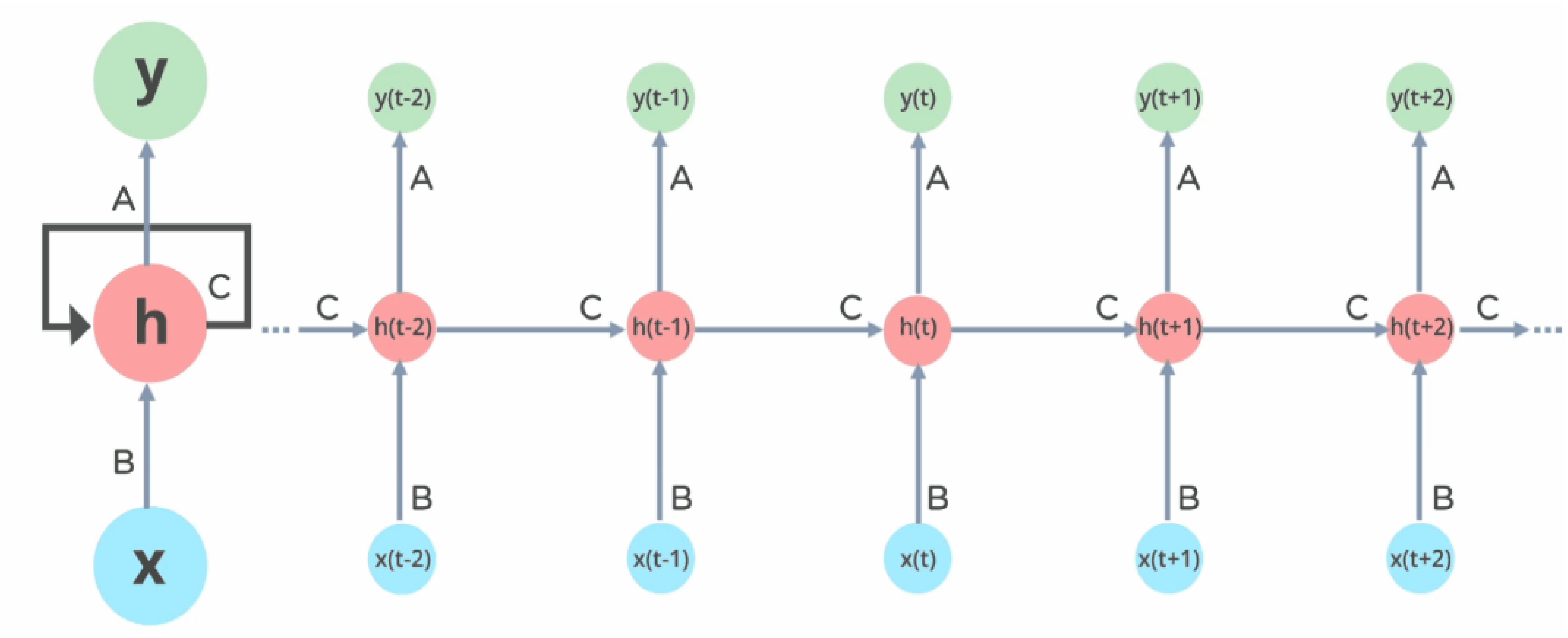
Source: Standford.edu

For each timestep t, the activation $a^{<t>}$ and the output $y^{<t>}$ are expressed as follows:

$$\boxed{a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a)} \quad \text{and} \quad \boxed{y^{<t>} = g_2(W_{ya}a^{<t>} + b_y)}$$

How does Recurrent Neural Networks work?

The information in recurrent neural networks cycles through a loop to the middle hidden layer.



RNN architecture can vary depending on the problem you're trying to solve. From those with a single input and output to those with many (with variations between).

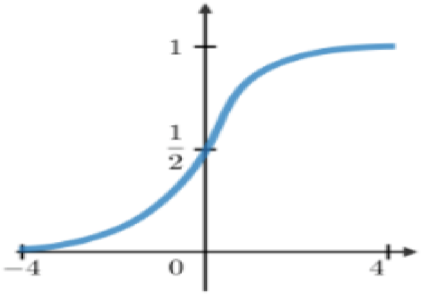
Below are some examples of RNN architectures that can help you better understand this.

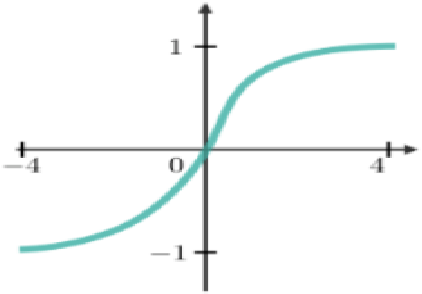
- **One To One:** There is only one pair here. A one-to-one architecture is used in traditional neural networks.
- **One To Many:** A single input in a one-to-many network might result in numerous outputs. One too many networks are used in the production of music, for example.
- **Many To One:** In this scenario, a single output is produced by combining many inputs from distinct time steps. Sentiment analysis and emotion identification use such networks, in which the class label is determined by a sequence of words.
- **Many To Many:** For many to many, there are numerous options. Two inputs yield three outputs. Machine translation systems, such as English to French or vice versa translation systems, use many to many networks.

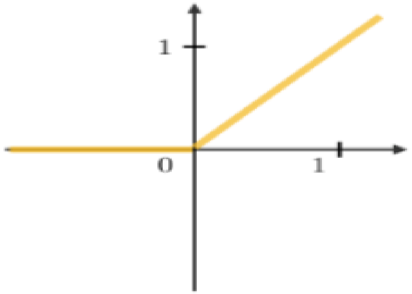
Common Activation Functions

A neuron's activation function dictates whether it should be turned on or off. Nonlinear functions usually transform a neuron's output to a number between 0 and 1 or -1 and 1.

Sigmoid	Tanh	RELU
$g(z) = \dfrac{1}{1+e^{-z}}$	$g(z) = \dfrac{e^{z}-e^{-z}}{e^{z}+e^{-z}}$	$g(z) = \max(0, z)$







The following are some of the most commonly utilized functions:

- **Sigmoid:** The formula **g(z) = 1/(1 + e^-z)** is used to express this.
- **Tanh:** The formula **g(z) = (e^-z – e^-z)/(e^-z + e^-z)** is used to express this.
- **Relu:** The formula **g(z) = max(0 , z)** is used to express this.

What is Cognitive Computing?

Cognitive Computing refers to individual technologies that perform specific tasks to facilitate **human intelligence**. Basically, these are smart decision support systems that we have been working with since the beginning of the internet boom. With recent breakthroughs in technology, these support systems simply use better data, better algorithms in order to get a better analysis of a huge amount of information.

Also, you can refer to Cognitive Computing as:

- **Understanding and simulating reasoning**
- **Understanding and simulating human behavior**

Using cognitive computing systems helps in making better human decisions at work. Some of the applications of cognitive computing include speech recognition, sentiment analysis, face detection, risk assessment, and fraud detection.

How Cognitive Computing Works?

Cognitive computing systems synthesize data from various information sources while weighing context and conflicting evidence to suggest suitable answers. To achieve this, cognitive systems include self-learning technologies using data mining, pattern recognition, and natural language processing (NLP) to understand the way the human brain works.

Using computer systems to solve problems that are supposed to be done by humans require huge structured and unstructured data. With time, cognitive systems learn to refine the way they identify patterns and the way they process data to become capable of anticipating new problems and model possible solutions.

To achieve these capabilities, cognitive computing systems must have some key attributes.

Key Attributes

- **Adaptive:** Cognitive systems must be flexible enough to understand the changes in the information. Also, the systems must be able to digest dynamic data in real-time and make adjustments as the data and environment change.

- **Interactive:** Human-computer interaction (HCI) is a critical component in cognitive systems. Users must be able to interact with cognitive machines and define their needs as those needs change. The technologies must also be able to interact with other processors, devices and cloud platforms.

- **Iterative and stateful:** Also, these systems must be able to identify problems by asking questions or pulling in additional data if the problem is incomplete. The systems do this by maintaining information about similar situations that have previously occurred.

- **Contextual:** Cognitive systems must understand, identify and mine contextual data, such as syntax, time, location, domain, requirements, a specific user's profile, tasks or goals. They may draw on multiple sources of information, including structured and unstructured data and visual, auditory or sensor data.

Cognitive Computing vs AI

The technologies behind Cognitive Computing are similar to the technologies behind AI. These include machine learning, deep learning, NLP, neural networks, etc. But they have various differences as well.

Cognitive Computing	Artificial Intelligence
Cognitive Computing focuses on **mimicking human behavior** and reasoning to solve complex problems.	AI **augments human thinking** to solve complex problems. It focuses on providing accurate results.
It **simulates** human thought processes to find solutions to complex problems.	AI **finds patterns** to learn or reveal hidden information and find solutions.
They simply **supplement information** for humans to make decisions.	AI is responsible for **making decisions** on their own minimizing the role of humans.
It is mostly used in sectors like **customer service, health care, industries**, etc.	It is mostly used in **finance, security, healthcare, retail, manufacturing**, etc.

Use case

Let us imagine a scenario where a person is deciding on a **career change**. An **AI assistant** will automatically assess the job seeker's **skills**, find a **relevant job** where his skills match the position, **negotiate pay** and benefits. And at the closing stage, it will inform the person that a decision has been made on his behalf.

Whereas, a cognitive assistant suggests **potential career paths** to the job seeker, besides furnishing the person with important details like additional **education requirements, salary comparison data**, and open job positions. However, in this case, the final decision must be still taken by the job seeker.