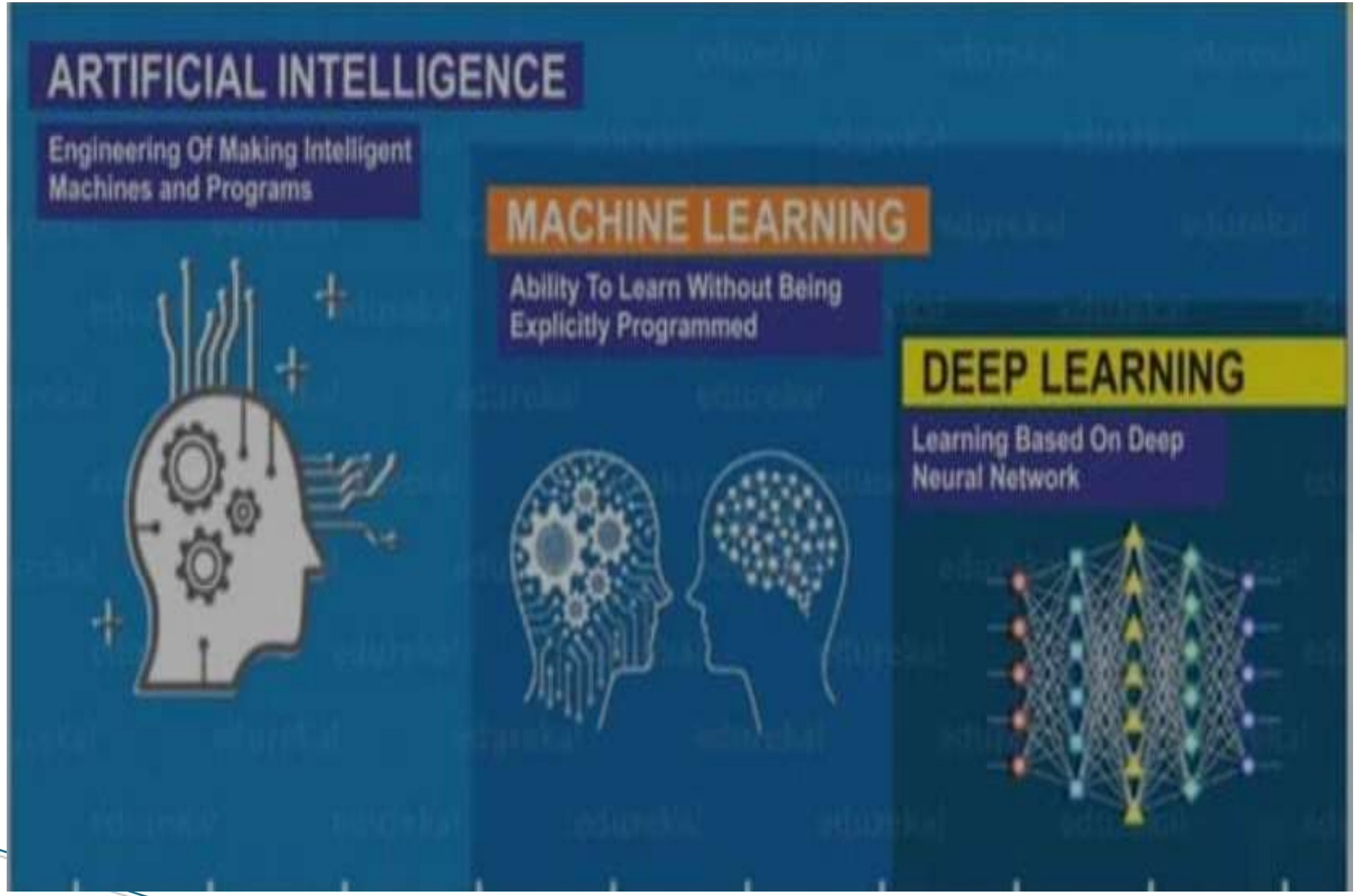
ARTIFICIAL INTELLIGENCE
CLASS-3

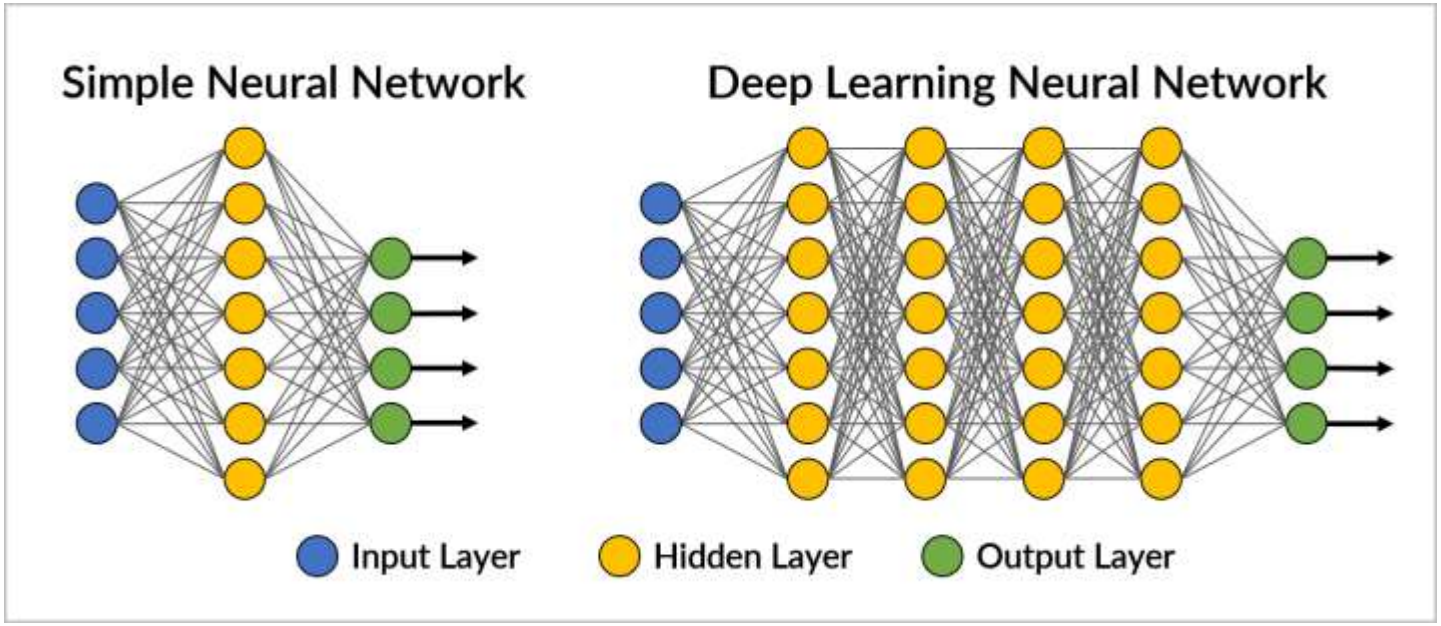
BY
GIRISH
KEERTHIVASAN

INTRODUCTION TO DEEP LEARNING

Deep Learning is a subfield of Machine Learning that involves the use of neural networks to model and solve complex problems. Neural networks are modeled after the structure and function of the human brain and consist of layers of interconnected nodes that process and transform data.


ARTIFICIAL INTELLIGENCE
Engineering Of Making Intelligent Machines and Programs
MACHINE LEARNING
Ability To Learn Without Being Explicitly Programmed
DEEP LEARNING
Learning Based On Deep Neural Network

DEEP LEARNING LAYERS



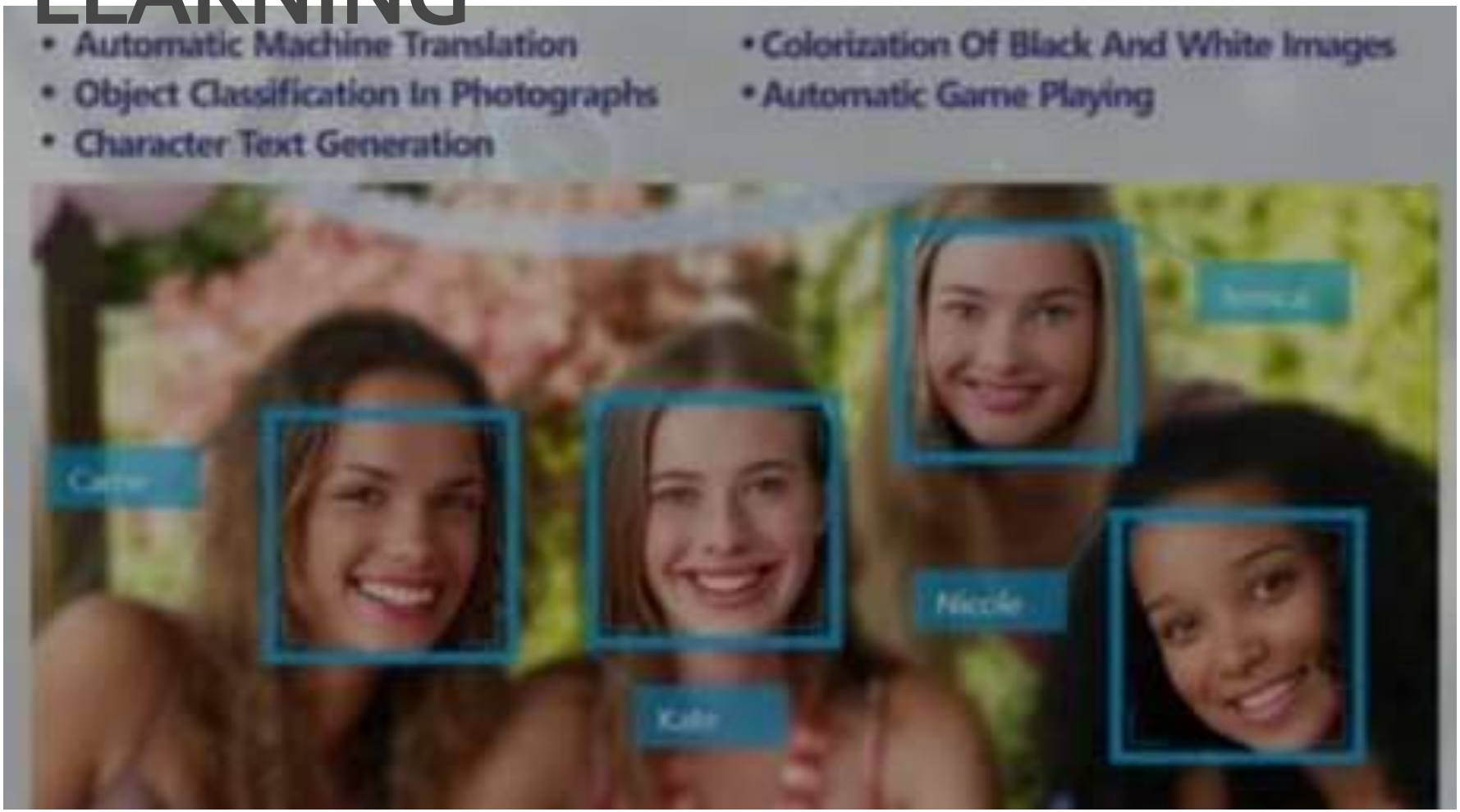
7 LAYERS

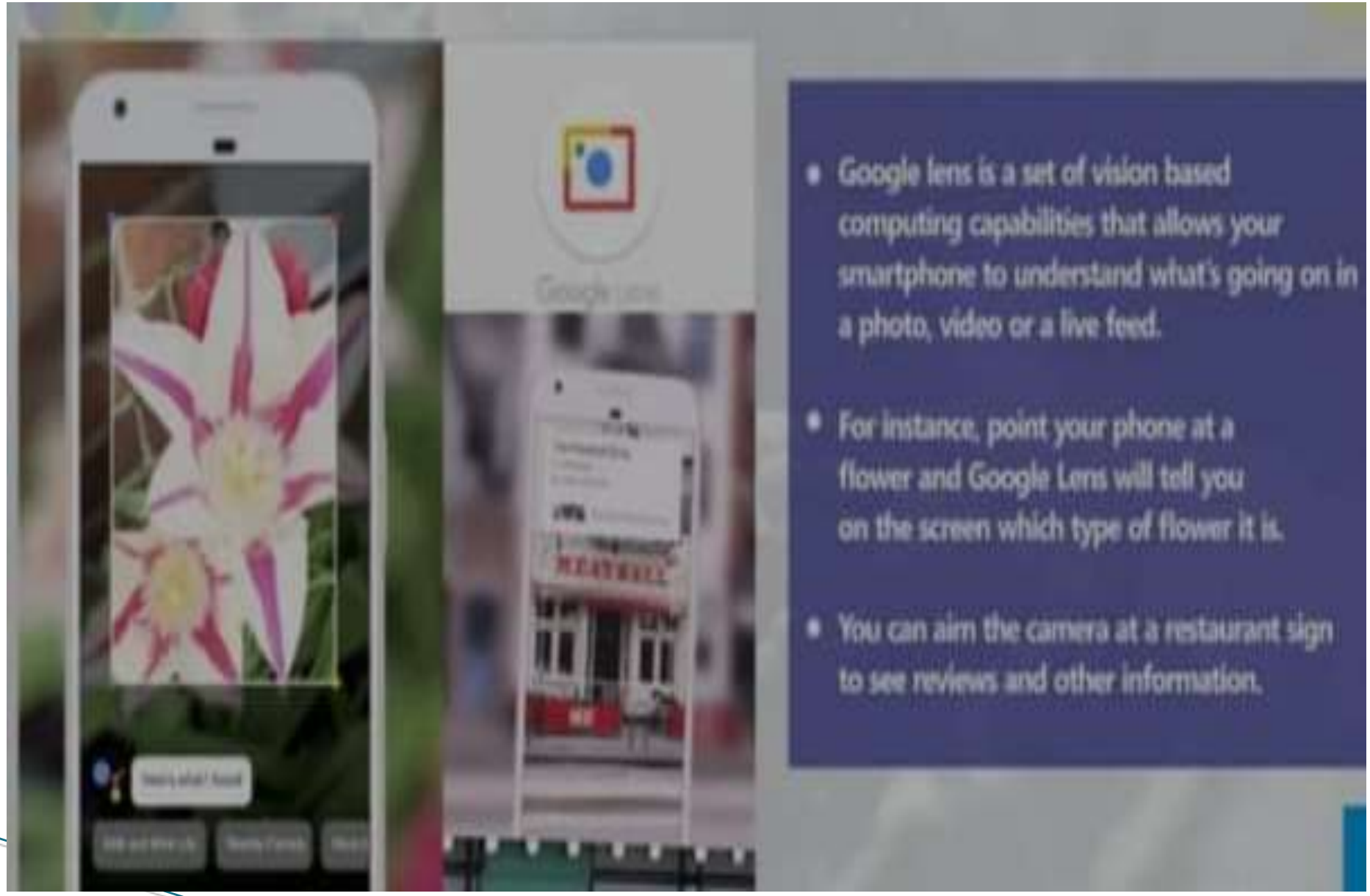
- Basic layer
- Convolution Layer
- Pooling layer
- Recurrent Layer
- Normalization Layer
- Regularization layer
- Attention layer

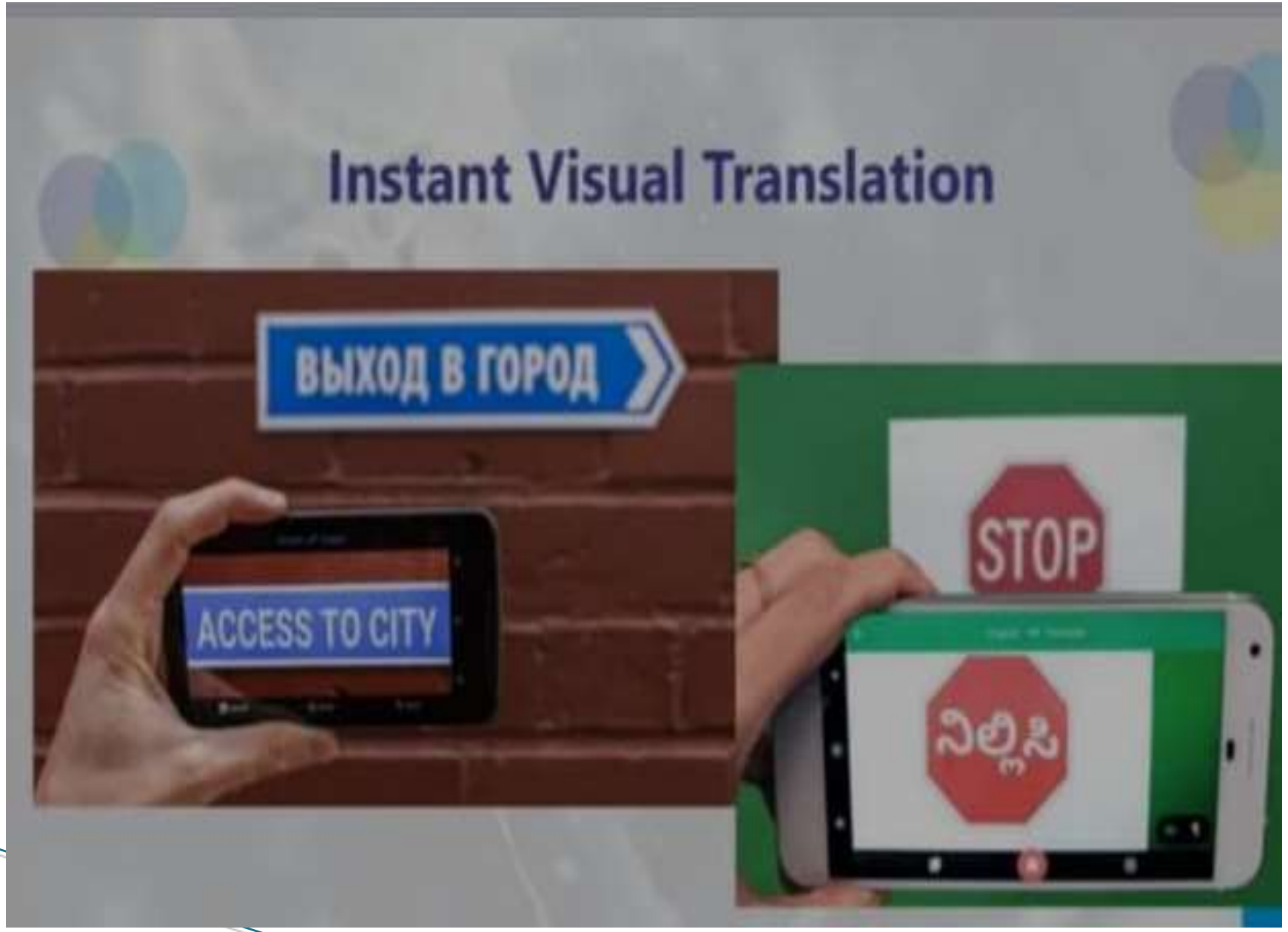
How Deep Learning Works?

- Deep Learning Is Implemented Through Neural Networks
- Motivation Behind Neural Networks Is The Biological Neuron.

APPLICATIONS OF DEEP LEARNING

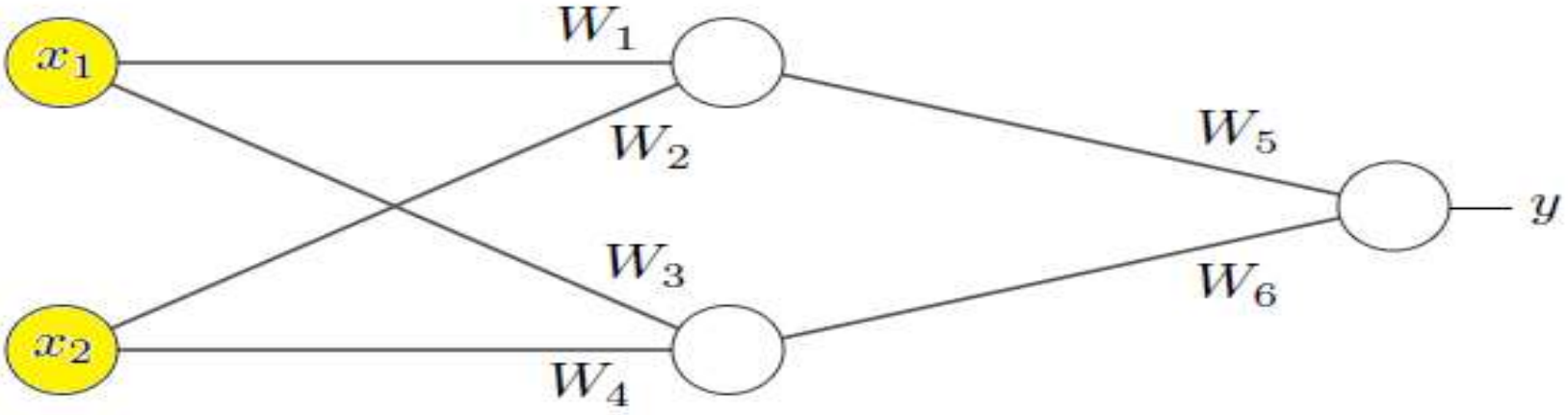



Google lens is a set of vision based computing capabilities that allows your smartphone to understand what's going on in a photo, video or a live feed.
For instance, point your phone at a flower and Google Lens will tell you on the screen which type of flower it is.
You can aim the camera at a restaurant sign to see reviews and other information.


Instant Visual Translation
ВЫХОД В ГОРОД
ACCESS TO CITY
STOP

ARTICIAL NEURAL NETWORK

Neural networks, also known as artificial neural networks (ANNs) or simulated neural networks (SNNs), are a subset of machine learning and are at the heart of deep learning algorithms. Their name and structure are inspired by the human brain, mimicking the way that biological neurons signal to one another.



X1,X2- INPUT
W1,W2,.....-WEIGHTS
Y-OUTPUT

Biological Neuron	Artificial Neuron
Dendrite	Inputs
Cell nucleus or Soma	Nodes
Synapses	Weights
Axon	Output

PYTHON IN GOOGLE COLAB

https://colab.research.google.com/ and sign in with your Google account

1) Create a New Notebook
2) Write and Run Python Code
3) Execute the Code

Introduction to **Python** Programming

- Python was developed by **Guido van Rossum** in late **1980**s at the **National Research Institute for Mathematics and Computer Science** in Netherlands.

• Guido named it after the Television show **"Monty Python's flying circus"**, of which Guido was, and presumably still is, a fan.

Features of Python Programming Language Cont'd

- **Object-Oriented Language**
  - Python supports object oriented language and concepts of classes and objects come into existence.
- **Extensible**
  - It implies that other languages such as C/C++ can be used to compile the code and thus it can be used further in our python code.
- **Large Standard Library**
  - Python has a large and broad library and prvides rich set of module and functions for rapid application development.
- **GUI Programming Support**
  - Graphical user interfaces can be developed using Python.
- **Integrated**
  - It can be easily integrated with languages like C, C++, JAVA etc.

Applications of Python

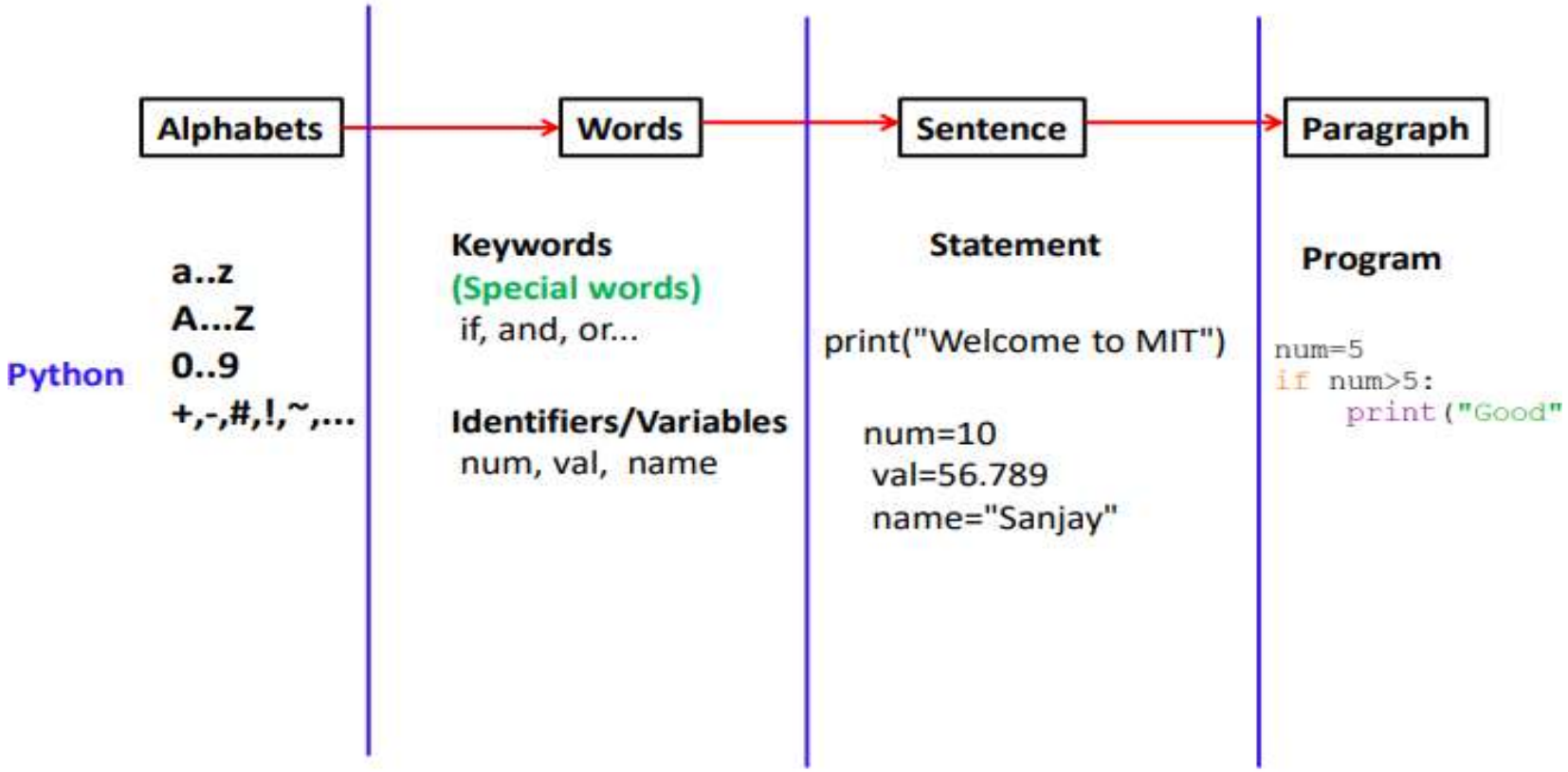
- Web development
- Machine Learning
- Data Analysis
- Scripting
- Game Development
- Embedded application
- Desktop Applications

Python installation

- Python installation is pretty simple, we can install it on any operating system such as Windows, Mac OS X, Ubuntu etc.
- To install the Python on your operating system, go to this link: **https://www.python.org/downloads/**. You will see a screen like this.

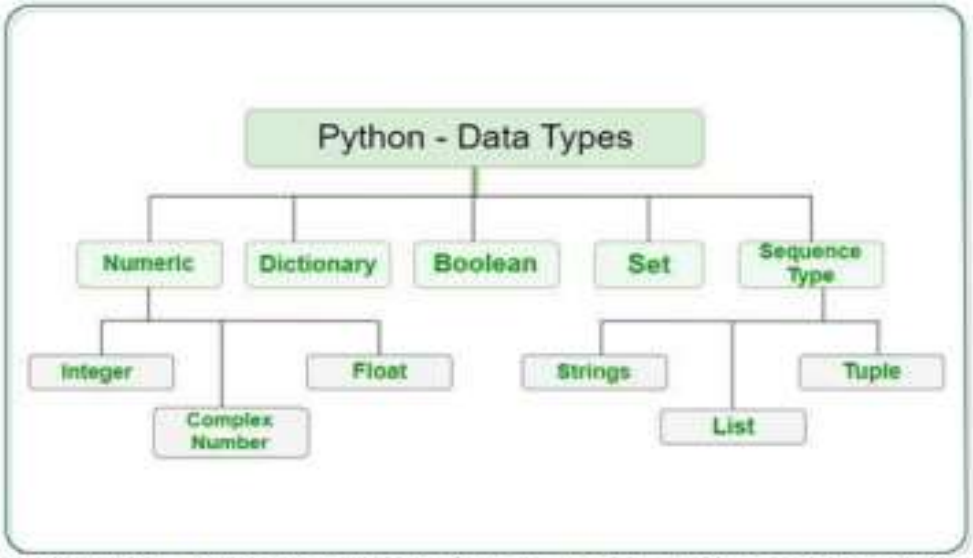

Python 3.9.2 (64-bit) Setup
Install Python 3.9.2 (64-bit)
Select Install Now to install Python with default settings, or choose
Customize to enable or disable features.
Install Now
C:\Users\Admin\AppData\Local\Programs\Python\Python39
Includes IDLE, pip and documentation
Creates shortcuts and file associations
Customize installation
Choose location and features
python
for
windows
Install launcher for all users (recommended)
Add Python 3.9 to PATH
Cancel

Language Basics



Python Data Types

- Data types are the **classification or categorization of data items**
- It is set of values and set of operators that may be applied to the values
- Python provides various standard data types



- Data types prevent the programmer from using values inappropriately.
- For example, it does not make sense to try to divide a string by two, 'Hello' / 2.
- The programmer knows this by common sense.
- Python knows it because 'Hello' belongs to the string data type, which does not include the division operation.

Numeric data type & Values

- **int**
  - Positive or negative **whole numbers** (without a fractional part) e.g. -10, 10, 456, 4654654.
- **float**
  - Any real number with a floating-point representation in which a **fractional component** is denoted by a decimal symbol or scientific notation e.g. 1.23, 3.4556789e2.
- **Complex**
  - A complex number contains an ordered pair, i.e., **x + iy** where x and y denote the real and imaginary parts, respectively. e.g. 2.14j, 2.0 + 2.3j.
- *Commas are never used in numeric literals.*

Numeric Literals		
Integer	**Float**	**Incorrect**
7	7. 7.0 7.123 0.0007 7000.567	7,000.123
3000	3500. 3500.0 3500.145	3,500 3,500.456
+3000	+3500. +3500.0 +3500.145	+3,500 +3,500.456
-3000	-3500. -3500.0 - 3500.145	-3,500 -3,500.456

PRACTICE SESSION

GOOGLE COLAB

Keywords in Python

- Python Keywords are **special reserved words** that convey a special meaning to the compiler/interpreter.

Keywords in Python programming language				
False	await	else	import	pass
None	break	except	in	raise
True	class	finally	is	return
and	continue	for	lambda	try
as	def	from	nonlocal	while
assert	del	global	not	with
async	elif	if	or	yield

Python Variables and Identifiers

- **Variable** is a name that is used to refer to memory location.
- Python variable is also known as an identifier and used to hold value.
- Variables are assigned values by use of the **assignment operator**, =.

  e.g. num = 10
- A variable can be assigned different values during a program's execution—hence, the name "variable."

  e.g. num=5

  num=10.5
- Wherever a variable appears in a program (except on the left-hand side of an assignment statement), *it is the value associated with the variable that is used*, and not the variable's name

  x=5

  x=x+4 means ,

  x=5+4 =9

Python Variables and Identifiers Cont'd

- **Some rules need to be followed for valid identifier naming:**
  - The first character of the variable must be an alphabet or underscore (_).
  - All the characters except the first character may be an alphabet of lower-case(a-z), upper-case (A-Z), underscore, or digit (0-9).
  - Identifier name must not contain any white-space, or special character (!, @, #, %, ^, &, *).
  - Identifier name must not be similar to any keyword defined in the language.
  - Identifier names are case-sensitive. For example, myname and myName are not the same. Identifiers can be of unlimited length.
  - Examples of valid identifiers: a1,speed,speed90 ,_n, F_name.
  - Examples of invalid identifiers: 1a, n%4, n 9, etc.
- Always give the identifiers a name that makes sense. While c = 10 is a valid name, writing count = 10 would make more sense, and it would be easier to figure out what it represents when you look at your code after a long gap.

Python Variables and Identifiers Cont'd

- **Object References:**
  - It is necessary to understand how the Python interpreter works when we declare a variable.
  - The process of treating variables is somewhat different from many other programming languages.
  - **Python is the highly object-oriented programming language;** that's why every data item belongs to a specific type of class.

```
>>> num=10
>>> type(num)
<class 'int'>
>>> f=34.78
>>> type(f)
<class 'float'>
>>> name="Ajay"
>>> type(name)
<class 'str'>
```

Arithmetic Operators

- Arithmetic operators are used to perform mathematical operations like addition, subtraction, multiplication, etc.

Operator	Meaning	Example
+	Unary plus or Add two operands	+x x+ y+ 2
-	Unary minus or Subtract right operand from the left	-x x - y- 2
*	Multiply two operands	x * y
/	**True division** - Divide left operand by the right one (always results into float)	x / y
%	**Modulus** - remainder of the division of left operand by the right	x % y (remainder of x/y)
//	**Floor division/Truncating division equivalent to math.floor(a/b)** - division that results into whole number adjusted to the left in the number line	x // y
**	**Exponentiation** - left operand raised to the power of right	x**y (x to the power y)

Arithmetic Operators Cont'd

Arithmetic Operators	Example	Result
+ (Addition)	10+25	35
- (Negation, Subtraction)	-10 10-25	-10 -15
*(Multiplication)	10*5	50
/(Division)	25/10	2.5
//(Truncated Division)	25//10 25//10.0	2 2.0
%(Modulus)	25%10	5
**(Exponentiation)	10**2	100

input() function Example

```
>>> num=input("Enter number:")
Enter number:25
>>> num
'25'
>>> num=int(input("Enter a number:"))
Enter a number:12
>>> num
12
>>> num=input("Enter name:")
Enter name:Sanjay
>>> num
'Sanjay'
>>> num=int(input("Enter name:"))
Enter name:Sanjay
Traceback (most recent call last):
  File "<pyshell#8>", line 1, in <module>
    num=int(input("Enter name:"))
ValueError: invalid literal for int() with base 10: 'Sanjay'
```

NEXT CLASS TOPICS

- MACHINE LEARNING PROCESS
- COMPARE BETWEEN ML AND DL
- DATA STRUCTURES
- LOOPS
- FUNCTIONS