Unit 5

Pattern Recognition

Introduction to pattern Recognition

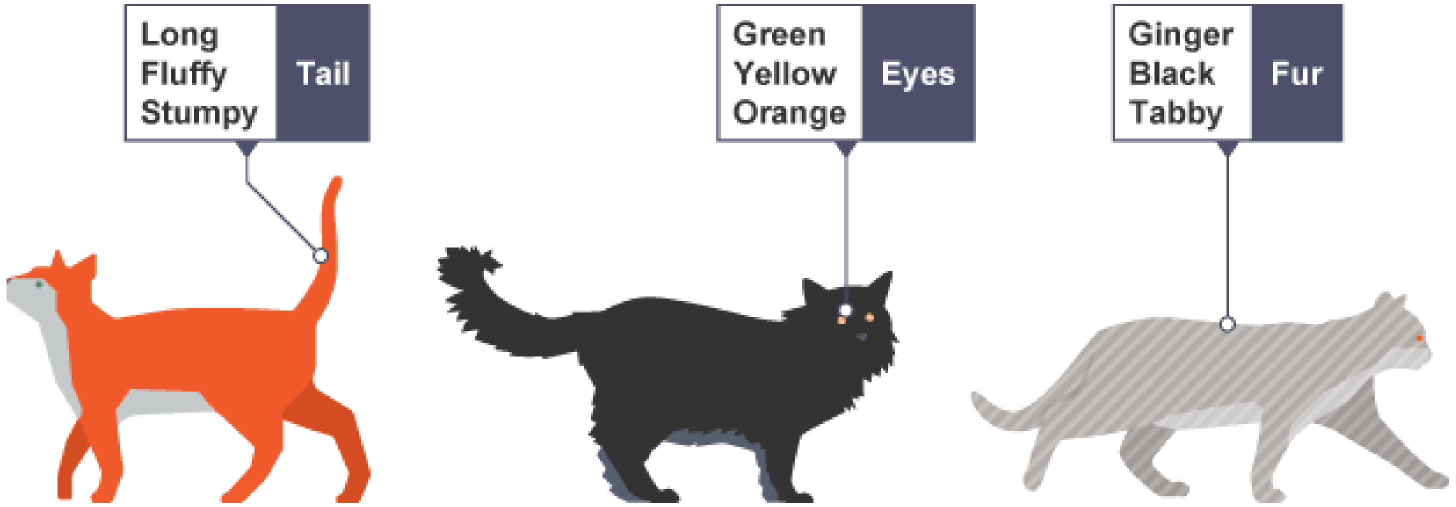
- **Pattern recognition** is the process of recognizing patterns by using a machine learning algorithm. Pattern recognition can be defined as the classification of data based on knowledge already gained or on statistical information extracted from patterns and/or their representation. One of the important aspects of pattern recognition is its application potential.
- **Examples:** Speech recognition, speaker identification, multimedia document recognition (MDR), automatic medical diagnosis.
- In a typical pattern recognition application, the raw data is processed and converted into a form that is amenable for a machine to use. Pattern recognition involves the classification and cluster of patterns.
    - In classification, an appropriate class label is assigned to a pattern based on an abstraction that is generated using a set of training patterns or domain knowledge. Classification is used in supervised learning.
    - Clustering generated a partition of the data which helps decision making, the specific decision-making activity of interest to us. Clustering is used in unsupervised learning.

All cats share common characteristics. Among other things **they all have eyes, tails and fur**. They also like to eat fish and make meowing sounds.

Because we know that all cats have eyes, tails and fur, we can make a good attempt at drawing a cat, simply by including these common characteristics.

In computational thinking, these characteristics are known as patterns. **Once we know how to describe one cat we can describe others, simply by following this pattern.** The only things that are different are the specifics:

- one cat may have green **eyes**, a long **tail** and black **fur**
- another cat may have yellow **eyes**, a short **tail** and striped **fur**



Patterns and Pattern Classes

A pattern is the arrangement of descriptors. The name *feature is* used often in the pattern recognition to denote a descriptor.

A *pattern class is a family* of patterns that share some common properties. Pattern classes are denoted ω1,ω2,…………,ωW, where W is the number of classes. Pattern recognition by machine involves techniques for assigning patterns to their respective classes automatically and with as little human intervention as possible. Three common pattern arrangements used in practice are vectors (for quantitative descriptions), strings and trees (for structural descriptions).

Pattern vectors are denoted by bold lowercase letters, such as **x**, **y**, and **z**, and represented as columns (that is, n x 1 matrices). Hence a pattern vector can be expressed in the form

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix}$$

Where each component, x_i, represents the i^{th} descriptor and n is the total number of such descriptors associated with the pattern.

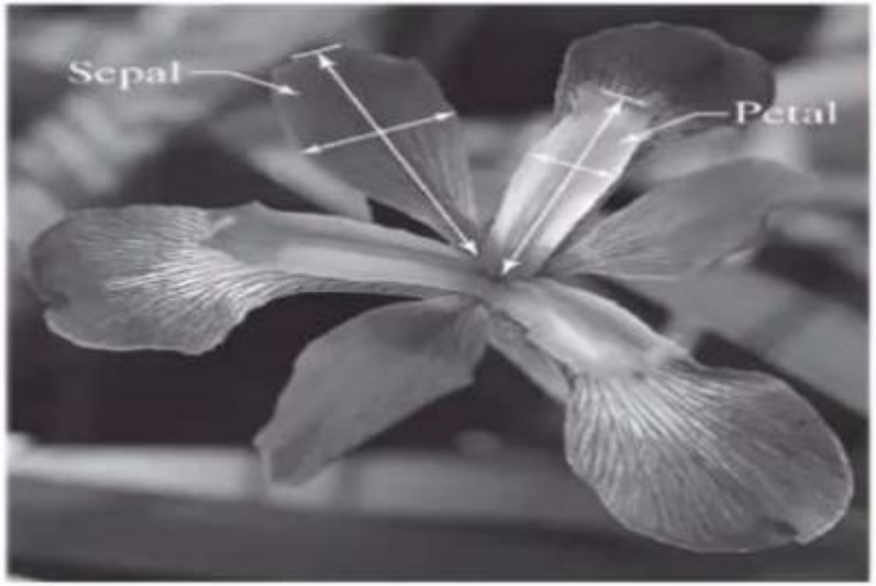
It can also express in the equivalent form

x = (x1, x2, , xn)^T

Where T indicates transposition of this matrix

The nature of the components of a pattern vector **x** depends on the approach used to describe the physical pattern itself.

Let us illustrate with an example that is both simple and gives a sense of history in the area of classification of measurements.



$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

x_1 = *Petal Width*

x_2 = *Petal Length*

x_3 = *Sepal Width*

x_4 = *Sepal Length*

Here in this vector, **x** is the pattern class and x_1, x_2, x_3, x_4 are different patterns in that class.

In our present terminology, each flower is described by two measurements, which leads to a 2-D pattern vector of the form

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

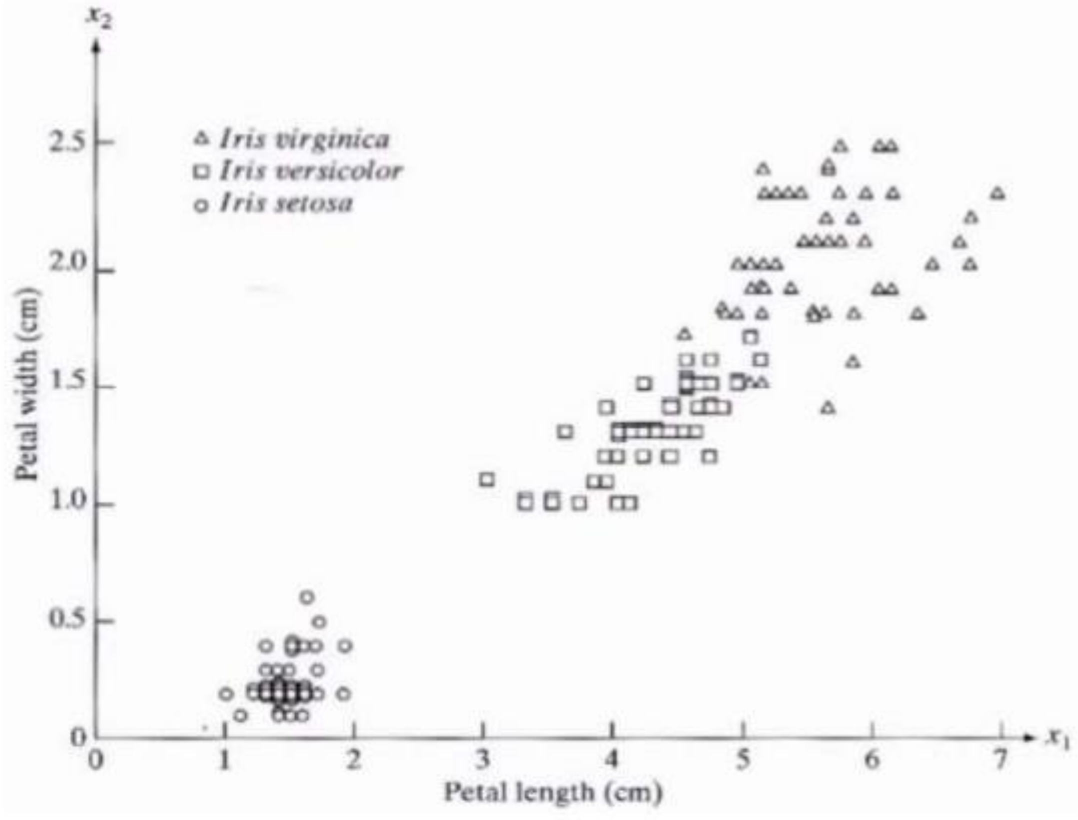
Where x_1 and x_2 correspond to petal length and width, respectively

Let's take some sample measurement for three types of flowers *setosa, virginica,* and *versicolor*. Then plot the two measurement petal length and petal width in graph and analyze the result.

Iris Versicolor

Iris Setosa

Iris Virginica



In this graph three types iris flower describe by two measurements

The three pattern classes in this case, denoted ω1, ω2, and ω3, correspond to the varieties *setosa, virginica,* and *versicolor, r*espectively. Because the petals of flowers vary in width and length, the pattern vectors describing these flowers also will vary, not only between different classes, but also within a class.

The above Figure shows length and width measurements for several samples of each type of iris. After a set of measurements has been selected (two in this case), the components of a pattern vector become the entire description of each physical sample.

Thus each flower in this case becomes a point in 2-D Euclidean space. We note also that measurements of petal width and length in this case adequately separated the class of *Iris setosa* from the other two but did not separate as successfully the *virginica* and *versicolor* types from each other. This result illustrates the classic *feature selection* problem, in which the degree of class seperability depends strongly on the choice of descriptors selected for an application.

Decision-Theoretic Methods

Decision theoretic approaches to recognition are based on the use of decision functions

$x = (x_1, x_1, x_1 \ldots\ldots x_n)$

It represents an n-dimensional pattern vector.

For W pattern classes ($\omega_1, \omega_2, \omega_3, \ldots \omega_w$), the basic problem in decision-theoretic pattern recognition is to find W decision functions $d_1(x), d_2(x), \ldots\ldots\ d_w(x)$ with the property that if a pattern x belongs to class ω_i then

$$d_i(x) > d_j(x)\ ,\ j=1,2,3\ldots\ldots W;\ j \neq i \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ 1$$

In another word an unknown pattern x is said to belongs to the i^{th} pattern class ω_i, if upon substitution of x into all decision functions, $d_i(x)$ yield the largest numerical value.

The decision boundary separating class ω_i from ω_j is given by the value of x for which

$d_i(x) = d_j(x)$ or, $d_i(x) - d_j(x) = 0$

Common practice is to identify the decision boundary between two classes by the single function

$$d_{ij}(x) = d_i(x) - d_j(x) = 0 \qquad \ldots\ldots\ldots\ldots\ldots 2$$

Thus $d_{ij}(x) > 0$ for patterns of class ω_i and $d_{ij}(x) < 0$ for pattern of class ω_j

Matching

Matching is the one of the popular decision theoretic method. Recognition techniques based on matching represent each class by a prototype pattern vector. An unknown pattern is assigned to the class to which, it is closest in terms of a predefined metric.

Minimum Distance Classifier

The simplest approach in matching method is the Minimum Distance Classifier, which, as its name implies, computes the (Euclidean) distance between the unknown and each of the prototype vectors. It chooses the smallest distance to make a decision.

Suppose that we define the prototype of each pattern class ω_j to be the mean vector of the patterns of that class:

$$m_j = \frac{1}{N_j}\sum_{x \epsilon \omega_j} x_j \qquad j = 1,2,\ldots\ldots\ldots\ldots\ldots,W \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ 3$$

Where N_j is the number of pattern vectors from class ω_j and the summation is taken over these vectors. W is the number of pattern classes.

One way to determine the class membership of an unknown pattern vector x is assign it to the class of its closest prototype.

If we use the Euclidean distance to determine similarity, the minimum distance classifier computes the distance as

$$d_j(x) = \|x-m_j\|,\ j = 1,2,\ldots\ldots,w \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ 4$$

Where, x is an unknown pattern vector, and m_j is mean vector of a class.

$\|x\| = (x^T x)^{1/2}$ is the Euclidean norm

Then assign x to class ω_j, if $d_j(x)$ is smallest distance. Hence, smallest distance implies to best match in this formulation.

Selecting the smallest distance is equivalent to evaluation of this equation

$$d_j(x) = \mathbf{x}^T\mathbf{m}_j - \frac{1}{2}\mathbf{m}_j^T\mathbf{m}_j \qquad j = 1,2,\ldots\ldots,W \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ 5$$

And assign x to class ω_j, if $d_j(x)$ yields the larges numerical value.

i.e.

$d_j(x) > d_i(x)$; $j = 1,2,\ldots\ldots,w$ and $j \neq i$

This formulation agrees the concept of decision function as defined in Equation 1.

***Note:** If there are two classes then boundary is perpendicular bisection line, if three classes then boundary is a plane like triangle, square and if more than there classes then boundary is hyperplane.*

Example 1: The two classes, Iris-versicolor and Iris-setosa, denoted ω_1 and ω_2 respectively, have sample mean vectors are

$m_1 = (4.3, 1.3)^T$, and $m_2 = (1.5, 0.3)^T$, find the equation of decision boundary.

Solution:

The decision function is

$d_1(x) = x^T m_1 - ½\, m_1^T m_1$

$= [x_1, x_2] \begin{bmatrix} 4.3 \\ 1.3 \end{bmatrix} - ½\, [4.3, 1.3] \begin{bmatrix} 4.3 \\ 1.3 \end{bmatrix}$

$= x_1\, 4.3 + x_2\, 1.3 - ½\, (4.3x4.3 + 1.3x1.3)$

$= x_1\, 4.3 + x_2\, 1.3 - ½\, (18.49 + 1.69)$

$= x_1\, 4.3 + x_2\, 1.3 - ½\, (20.18)$

$= 4.3x_1 + 1.3x_2 - 10.1$

$d_2(x) = x^T m_2 - ½\, m_2^T m_2$

$= 1.5x_1 + 0.3x_2 - 1.17$

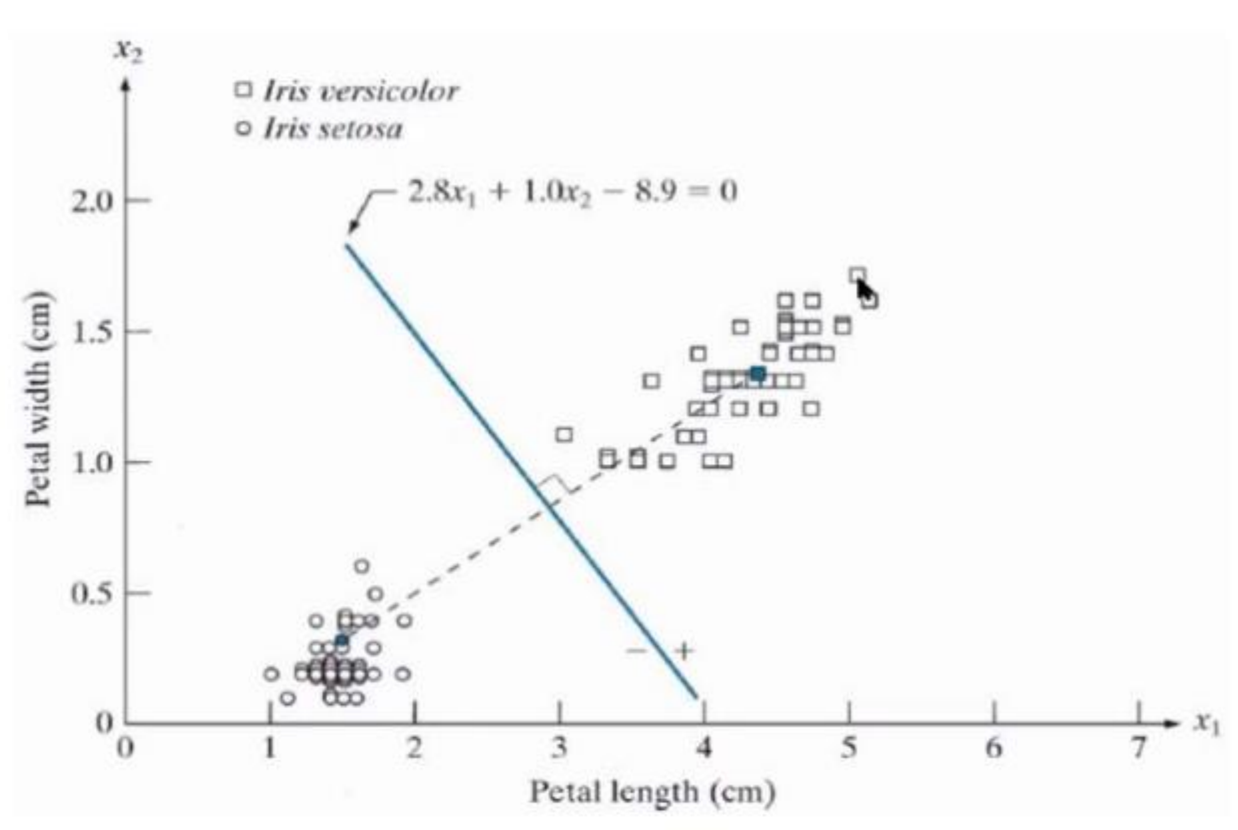
The Equation of decision boundary is

$D_{12}(x) = d_1(x) - d_2(x) = 0$

$= 4.3x_1 + 1.3x_2 - 10.1 - (1.5x_1 + 0.3x_2 - 1.17)$

$= 4.3x_1 + 1.3x_2 - 10.1 - 1.5x_1 - 0.3x_2 + 1.17$

$= 2.8x_1 + 1.0x_2 - 8.9 = 0$



Example 2: The two classes, Iris-versicolor and Iris-setosa, denoted ω_1 and ω_2 respectively, have sample mean vectors are $m_1 = (4.3, 1.3)^T$, and $m_2 = (1.5, 0.3)^T$. An unknown pattern P with vector $P = (4.4, 1.5)^T$ is found, find in which class pattern P belongs?

Solution:

Given, mean vector of two classes are

$m_1 = (4.3, 1.3)^T$

$m_2 = (1.5, 0.3)^T$

Unknown pattern is

$P = (4.4, 1.5)^T$

We know the decision function is

$d_n(x) = x^T m_n - ½\, m_n^T m_n$

For first class i.e. $m_1 = (4.3, 1.3)^T$

$d_1(p) = p^T m_1 - ½\, m_1^T m_1$

$= [4.4, 1.5] \begin{bmatrix} 4.3 \\ 1.3 \end{bmatrix} - ½\, [4.3, 1.3] \begin{bmatrix} 4.3 \\ 1.3 \end{bmatrix}$

$= 4.4x4.3 + 1.5x1.3 - ½\, (4.3x4.3 + 1.3x1.3)$

$= 18.9 + 1.9 - ½\, (18.49 + 1.69)$

$= 35.9 - ½\, (20.18)$

$= 35.9 - 10.1$

$= 25.8$

For second class i.e. $m_2 = (1.5, 0.3)^T$

$d_2(p) = p^T m_2 - ½\, m_2^T m_2$

$= [4.4, 1.5] \begin{bmatrix} 1.5 \\ 0.3 \end{bmatrix} - ½\, [1.5, 0.3] \begin{bmatrix} 1.5 \\ 0.3 \end{bmatrix}$

$= 4.4x1.5 + 1.5x0.3 - ½\, (1.5x1.5 + 0.3x0.3)$

$= 6.6 + 0.45 - ½\, (2.25 + 0.09)$

$= 7.05 - ½\, (2.34)$

$= 7.05 - 1.17$

$= 5.88$

Now the $d_1(P) > d_2(P)$

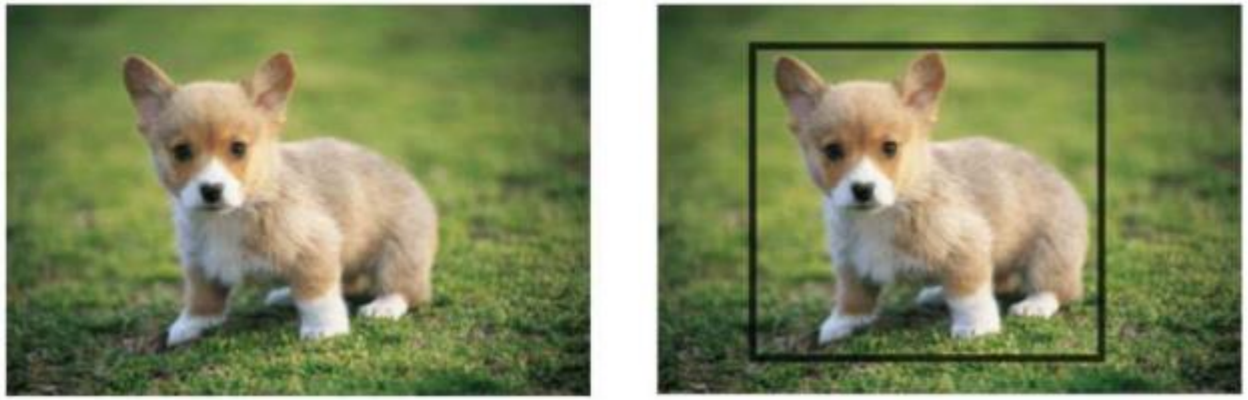
Hence, Pattern P is belongs to first class i.e. Iris-versicolor.

Neural Network Based Image Recognition

Image recognition

Image recognition (or image classification) is the task of identifying images and categorizing them in one of several predefined distinct classes. So, image recognition software and apps can define what's depicted in a picture and distinguish one object from another. The field of study aimed at enabling machines with this ability is called computer vision. Being one of the computer vision (CV) tasks, image classification serves as the foundation for solving different CV problems, including:

Image classification with localization: placing an image in a given class and drawing a bounding box around an object to show where it's located in an image.



Object detection: categorizing multiple different objects in the image and showing the location of each of them with bounding boxes. So, it's a variation of the image classification with localization tasks for numerous objects.

Object Detection

Object (semantic) segmentation: identifying specific pixels belonging to each object in an image instead of drawing bounding boxes around each object as in object detection.

Semantic Segmentation

Instance segmentation: differentiating multiple objects (instances) belonging to the same class (each person in a group).

Instance Segmentation

Structure for performing Image Classification

1.Image Pre-processing*:* The aim of this process is to improve the image data (features) by suppressing unwanted distortions and enhancement of some important image features so that the computer vision models can benefit from this improved data to work on. Steps for image pre-processing includes Reading image, Resizing image, and Data Augmentation (Gray scaling of image, Reflection, Gaussian Blurring, Histogram, Equalization, Rotation, and Translation).

2.Detection of an object*:* Detection refers to the localization of an object which means the segmentation of the image and identifying the position of the object of interest.

3.Feature extraction and training*:* This is a crucial step wherein statistical or deep learning methods are used to identify the most interesting patterns of the image, features that might be unique to a particular class and that will, later on, help the model to differentiate between different classes. This process where the model learns the features from the dataset is called model training.

4.Classification of the object*:* This step categorizes detected objects into predefined classes by using a suitable classification technique that compares the image patterns with the target patterns.

Neural Network

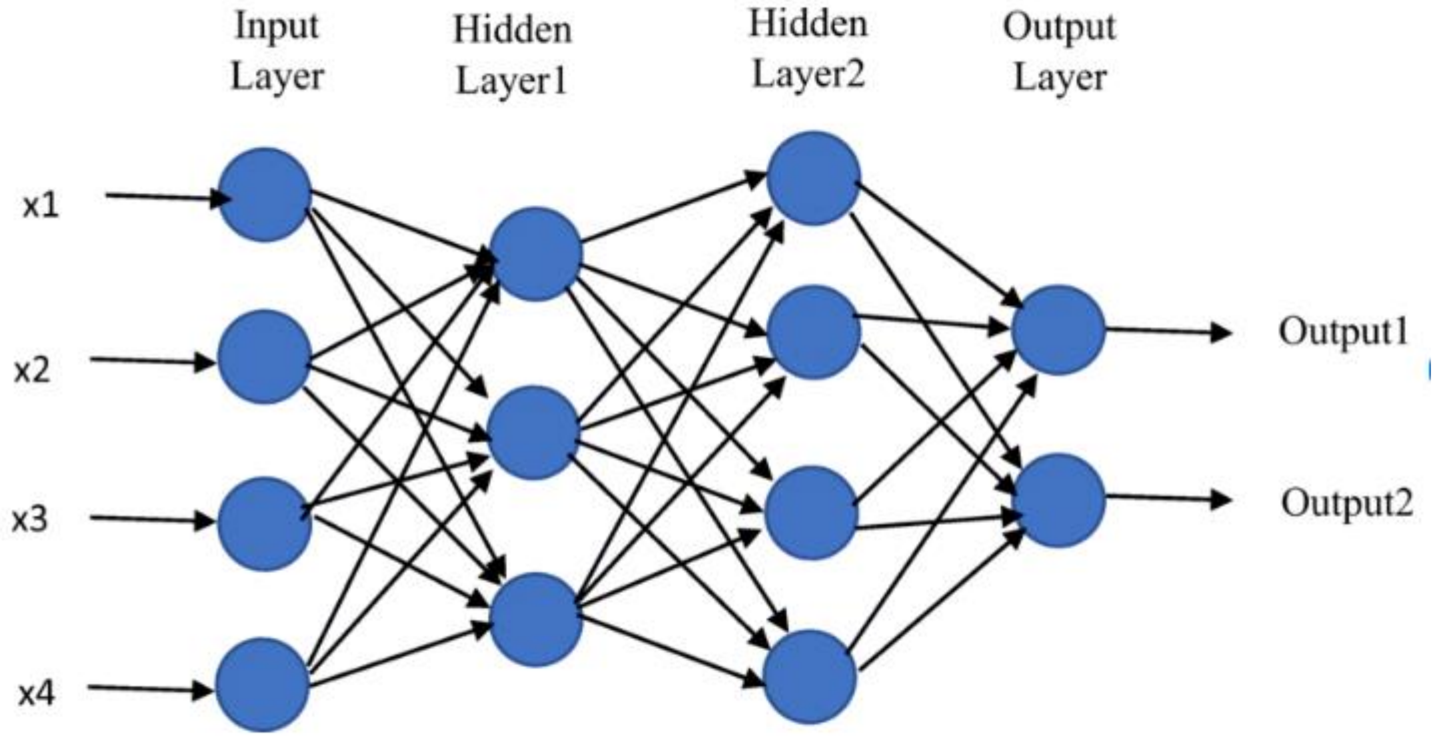
Researchers can use deep learning models for solving computer vision tasks. Deep learning is a machine learning technique that focuses on teaching machines to learn by example. Since most deep learning methods use neural network architectures, deep learning models are frequently called deep neural networks.

Image recognition is one of the tasks in which deep neural networks (DNNs). Neural networks are computing systems designed to recognize patterns. Their architecture is inspired by the human brain structure, hence the name.

They consist of three types of layers:

- Input Layers
- Hidden Layers
- Output Layers

The input layer receives a signal, the hidden layer processes it, and the output layer makes a decision or a forecast about the input data. Each network layer consists of interconnected nodes (artificial neurons) that do the computation. While traditional neural networks have up to three hidden layers, deep networks may contain hundreds of them.



General architecture of artificial neural network

Convolutional Neural Network

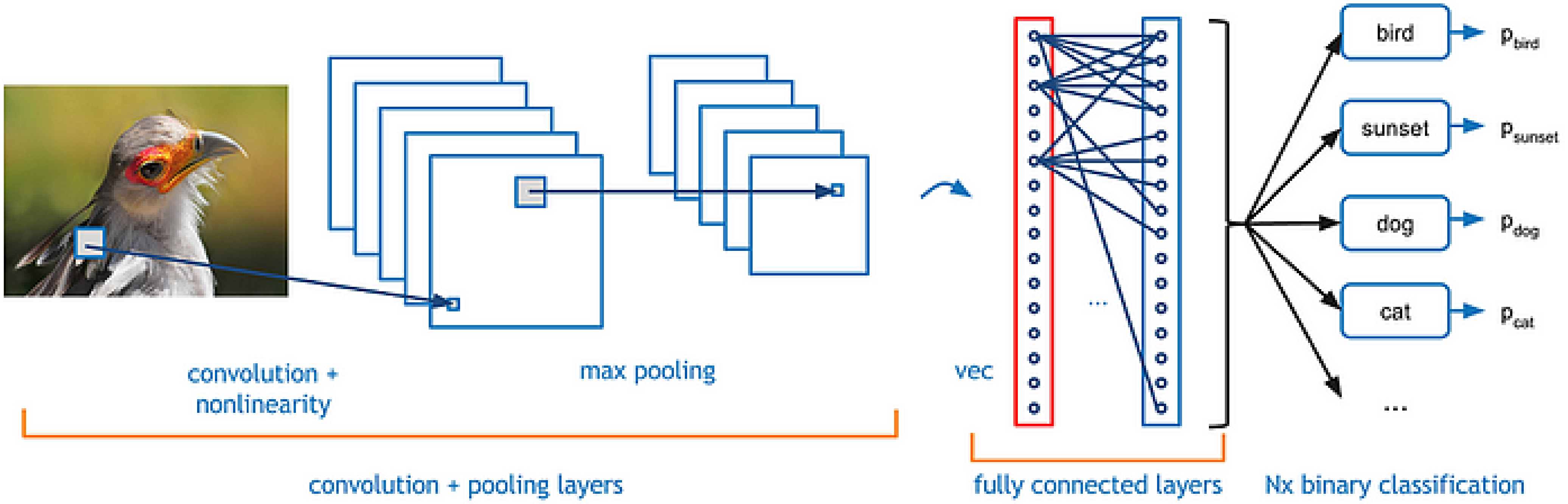
- Convolutional Neural Network (CNN, or ConvNet) are a special kind of multi-layer neural networks, designed to recognize visual patterns directly from pixel images with minimal pre-processing.

- It is a special architecture of artificial neural networks. Convolutional neural network uses some of its features of visual cortex and have therefore achieved state of the art results in computer vision tasks.

- Convolutional neural networks are comprised of two very simple elements, namely convolutional layers and pooling layers. Although simple, there are near-infinite ways to arrange these layers for a given computer vision problem.

- The elements of a convolutional neural network, such as convolutional and pooling layers, are relatively straightforward to understand. The challenging part of using convolutional neural networks in practice is how to design model architectures that best use these simple elements.

- The reason why convolutional neural network is hugely popular is because of their architecture, the best thing is there is no need of feature extraction. The system learns to do feature extraction and the core concept is, it uses convolution of image and filters to generate invariant features which are passed on to the next layer. The features in next layer are convoluted with different filters to generate more invariant and abstract features and the process continues till it gets final feature/output which is invariant to occlusions.


convolution +
nonlinearity
max pooling
convolution + pooling layers
vec
...
fully connected layers
bird
sunset
dog
cat
...
p_{bird}
p_{sunset}
p_{dog}
p_{cat}
Nx binary classification