Artificial Neural Networks

Biological Inspirations

Humans perform complex tasks like vision, motor control, or language understanding very well.

One way to build intelligent machines is to try to imitate the (organizational principles of) human brain.

Human Brain

- The brain is a highly complex, non-linear, and parallel computer, composed of some 10^{11} neurons that are densely connected ($\sim 10^4$ connection per neuron). *We have just begun to understand how the brain works...*

- A neuron is much slower (10^{-3}sec) compared to a silicon logic gate (10^{-9}sec), however the massive interconnection between neurons make up for the comparably slow rate.
  - Complex perceptual decisions are arrived at quickly (within a few hundred milliseconds)

- 100-Steps rule: Since individual neurons operate in a few milliseconds, calculations do not involve more than about 100 serial steps and the information sent from one neuron to another is very small (a few bits)

- Plasticity: Some of the neural structure of the brain is present at birth, while other parts are developed through learning, especially in early stages of life, to adapt to the environment (new inputs).

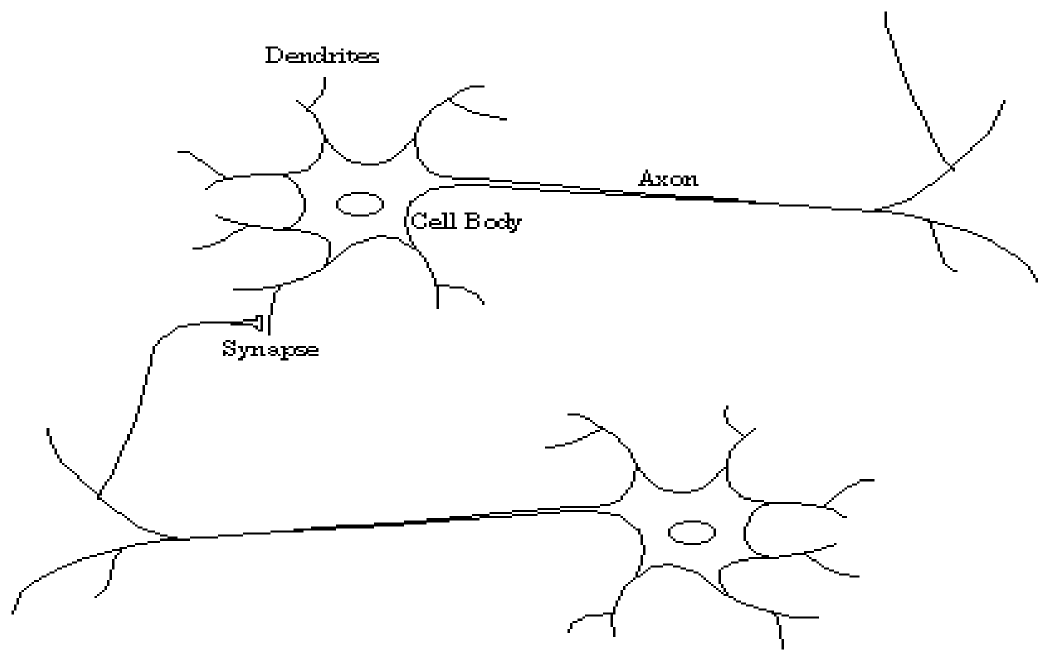
Biological Neuron

A variety of different neurons exist (motor neuron, on-center off-surround visual cells…), with different branching structures.

The connections of the network and the strengths of the individual synapses establish the function of the network.

Biological Neuron

- dendrites: nerve fibres carrying electrical signals to the cell
- cell body: computes a non-linear function of its inputs
- axon: single long fiber that carries the electrical signal from the cell body to other neurons
- synapse: the point of contact between the axon of one cell and the dendrite of another, regulating a chemical connection whose strength affects the input to the cell.



Artificial Neural Networks

Computational models inspired by the human brain:

– Massively parallel, distributed system, made up of simple processing units (neurons)

– Synaptic connection strengths among neurons are used to store the acquired knowledge.

– Knowledge is acquired by the network from its environment through a learning process

Properties of ANNs

Learning from examples
- labeled or unlabeled

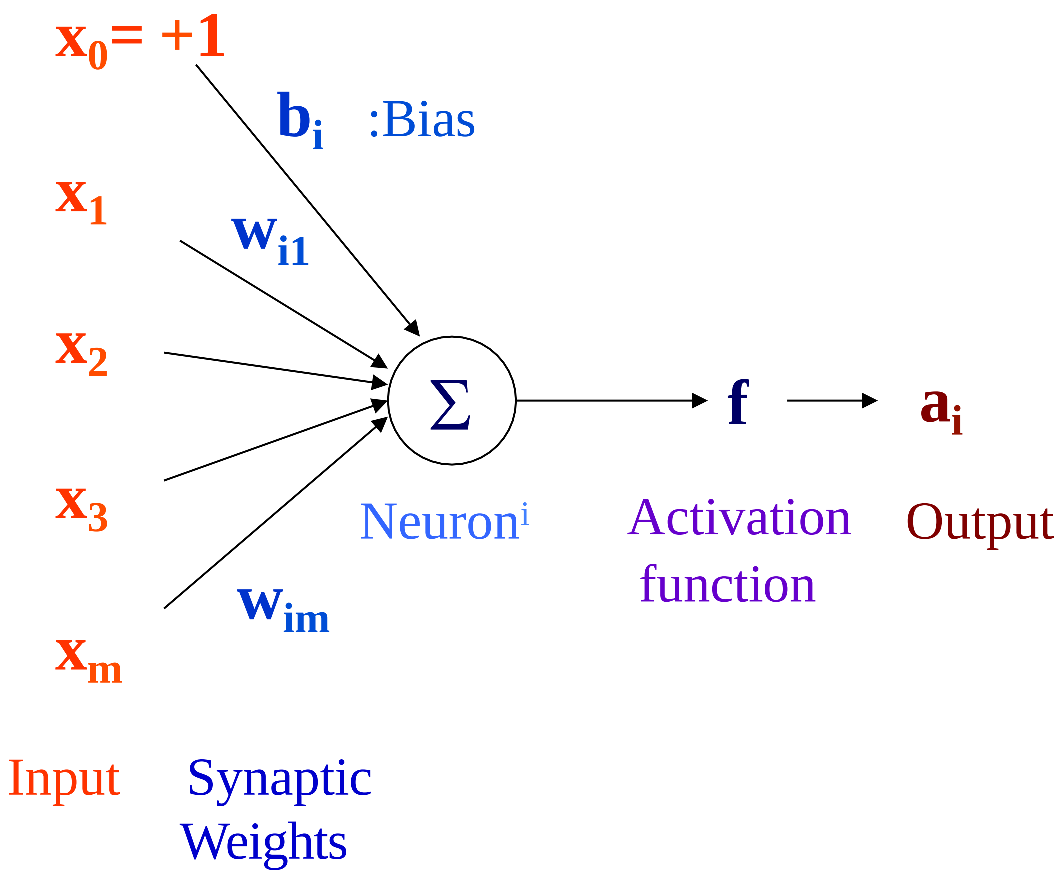
Adaptivity
- changing the connection strengths to learn things

Non-linearity
- the non-linear activation functions are essential

Fault tolerance
- if one of the neurons or connections is damaged, the whole network still works quite well

Thus, they might be better alternatives than classical solutions for problems characterised by:
- high dimensionality, noisy, imprecise or imperfect data; and
- a lack of a clearly stated mathematical solution or algorithm

Artificial Neuron Model



Bias

$$a_i = f(n_i) = f\left(\sum_{j=1}^{n} w_{ij}x_j + b_i\right)$$

An artificial neuron:

- computes the weighted sum of its input (called its **net input**)
- adds its bias
- passes this value through an activation function

We say that the neuron "fires" (i.e. becomes active) if its output is above zero.

Bias

Bias can be incorporated as another weight clamped to a fixed input of +1.0

This extra free variable (bias) makes the neuron more powerful.

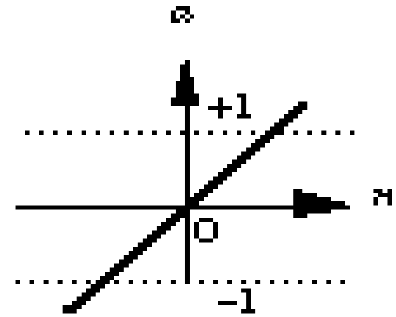
$$a_i = f(n_i) = f\left(\sum_{j=\mathbf{0}}^{n} w_{ij}x_j\right) = f(\mathbf{w}_i.\mathbf{x}_j)$$

Activation functions

Also called the squashing function as it limits the amplitude of the output of the neuron.

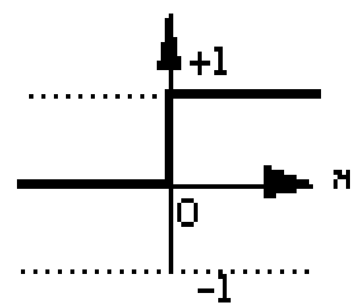
Many types of activations functions are used:

- linear: $a = f(n) = n$

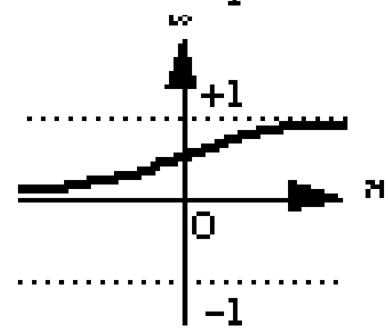


- threshold: (hardlimiting) $a = \{1$ if $n >= 0$

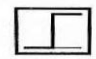
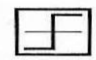
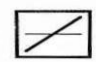
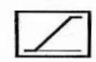
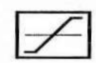
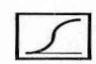
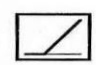
  0 if $n < 0$



- sigmoid: $a = 1/(1+e^{-n})$



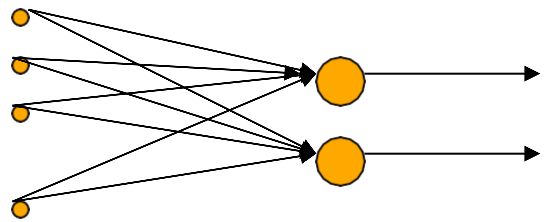
Activation Functions

Name	Input/Output Relation	Icon	MATLAB Function
Hard Limit	$a = 0 \quad n < 0$ $a = 1 \quad n \geq 0$		hardlim
Symmetrical Hard Limit	$a = -1 \quad n < 0$ $a = +1 \quad n \geq 0$		hardlims
Linear	$a = n$		purelin
Saturating Linear	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n \leq 1$ $a = 1 \quad n > 1$		satlin
Symmetric Saturating Linear	$a = -1 \quad n < -1$ $a = n \quad -1 \leq n \leq 1$ $a = 1 \quad n > 1$		satlins
Log-Sigmoid	$a = \dfrac{1}{1 + e^{-n}}$		logsig
Hyperbolic Tangent Sigmoid	$a = \dfrac{e^{n} - e^{-n}}{e^{n} + e^{-n}}$		tansig
Positive Linear	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n$		poslin
Competitive	$a = 1$ neuron with max n $a = 0$ all other neurons	C	compet

Different Network Topologies

Single layer feed-forward networks

– Input layer projecting into the output layer



Input layer

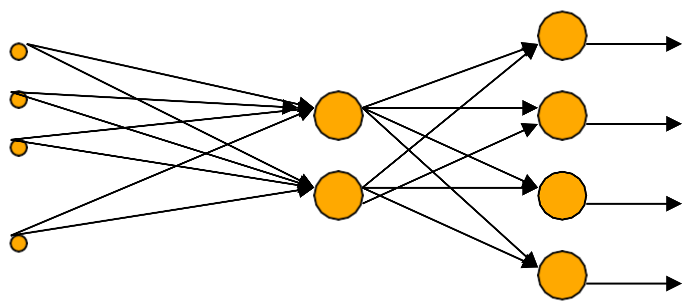
Output layer

Different Network Topologies

Multi-layer feed-forward networks

- One or more hidden layers.
- Input projects only from previous layers onto a layer.

  typically, only from one layer to the next



Input layer

Hidden layer

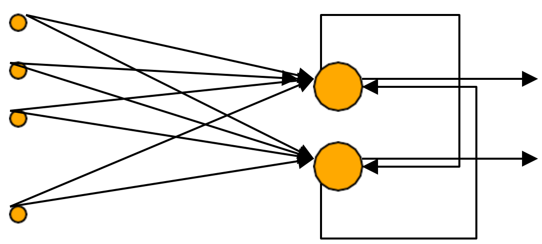
Output layer

2-layer or 1-hidden layer *fully connected* network

Different Network Topologies

Recurrent networks

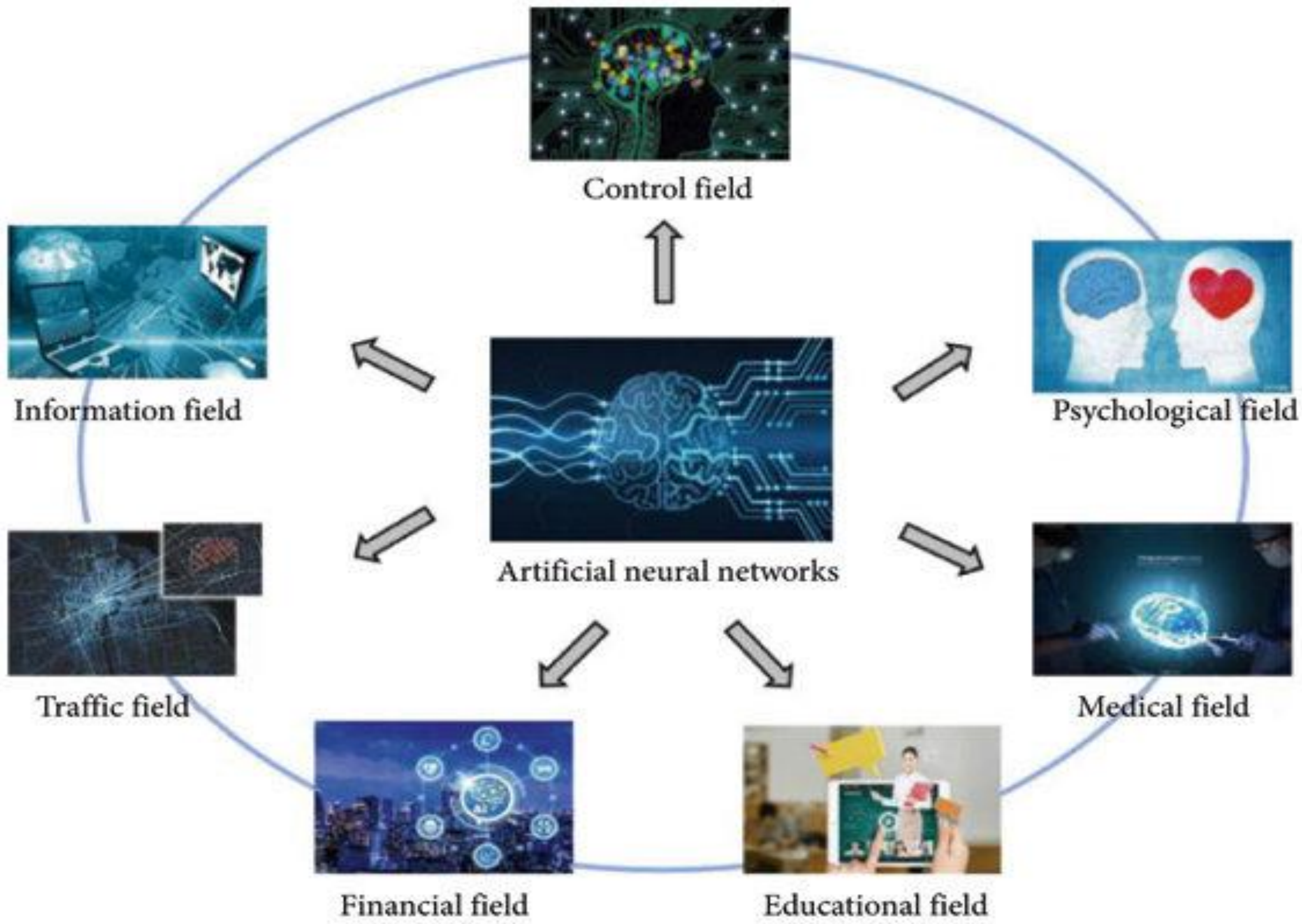
– A network with feedback, where some of its inputs are connected to some of its outputs (discrete time).



Input layer

Output layer

Applications of ANNs



Artificial Neural Networks

Early ANN Models:

- Perceptron, ADALINE, Hopfield Network

Current Models:

- Deep Learning Architectures
- Multilayer feedforward networks (Multilayer perceptrons)
- Radial Basis Function networks
- Self Organizing Networks

How to Decide on a Network Topology?

- # of input nodes?
  - Number of features
- # of output nodes?
  - Suitable to encode the output representation
- transfer function?
  - Suitable to the problem
- # of hidden nodes?
  - Not exactly known

Advantages of ANN

- Artificial neural networks have the ability to provide the data to be processed in parallel, which means they can handle more than one task at the same time.
- Artificial neural networks have been in resistance. This means that the loss of one or more cells, or neural networks, influences the performance of Artificial Neural networks.
- Artificial neural networks are used to store information on the network so that, even in the absence of a data pair, it does not mean that the network is not generating results.
- Artificial neural networks are gradually being broken down, which means that they will not suddenly stop working and these networks are gradually being broken down.
- We are able to train ANN’s that these networks learn from past events and make decisions.

Disadvantages of ANN

- As we mentioned before, with ANN arms hanging along with the execution of parallel processing, and so they need processors that support parallel processing, so the ANNs are dependent on the hardware.
- Since it's similar to the functionality of the human brain, we may not be able to determine what is the proper network structure of an Artificial Neural network.
- Not only do artificial neural networks, but also the statistical models can be trained with only numeric data, so it makes it very difficult for ANN to understand the problem statement.
- When an artificial neural network that provides a solution to the problem statements that we really don't know on what basis it will give the solution, and this time, ANN is not a reliable

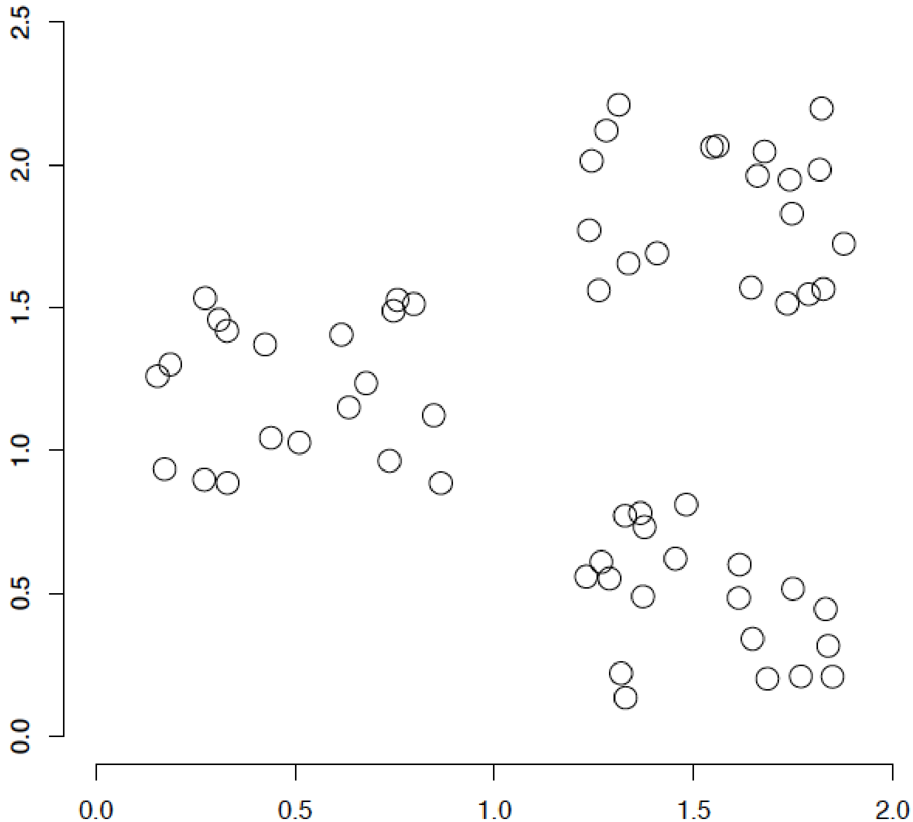
“Thank You”

Clustering

What is clustering?

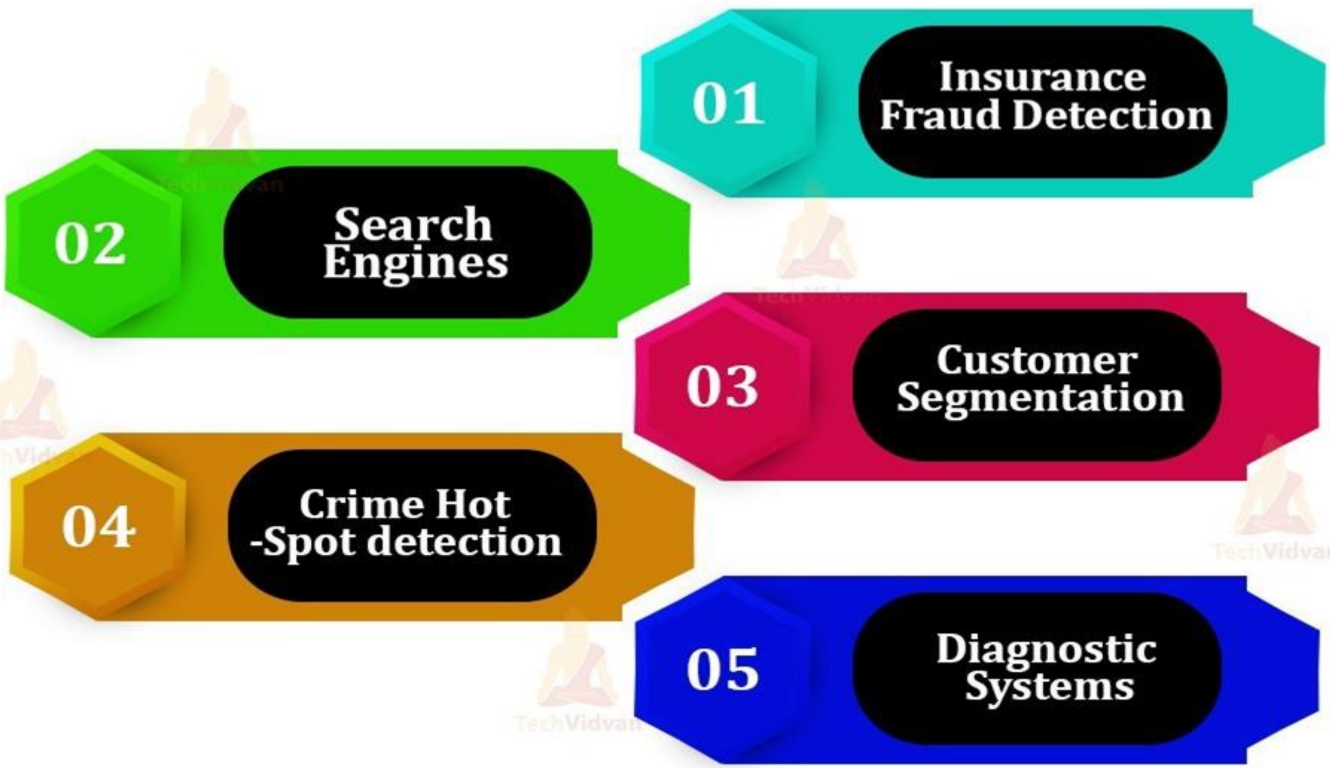
- Clustering: the process of grouping a set of objects into classes of similar objects
  - Documents within a cluster should be similar.
  - Documents from different clusters should be dissimilar.
- The commonest form of *unsupervised learning*
    - Unsupervised learning = learning from raw data, as opposed to supervised data where a classification of examples is given
  - A common and important task that finds many applications in IR and other places

A data set with clear cluster structure

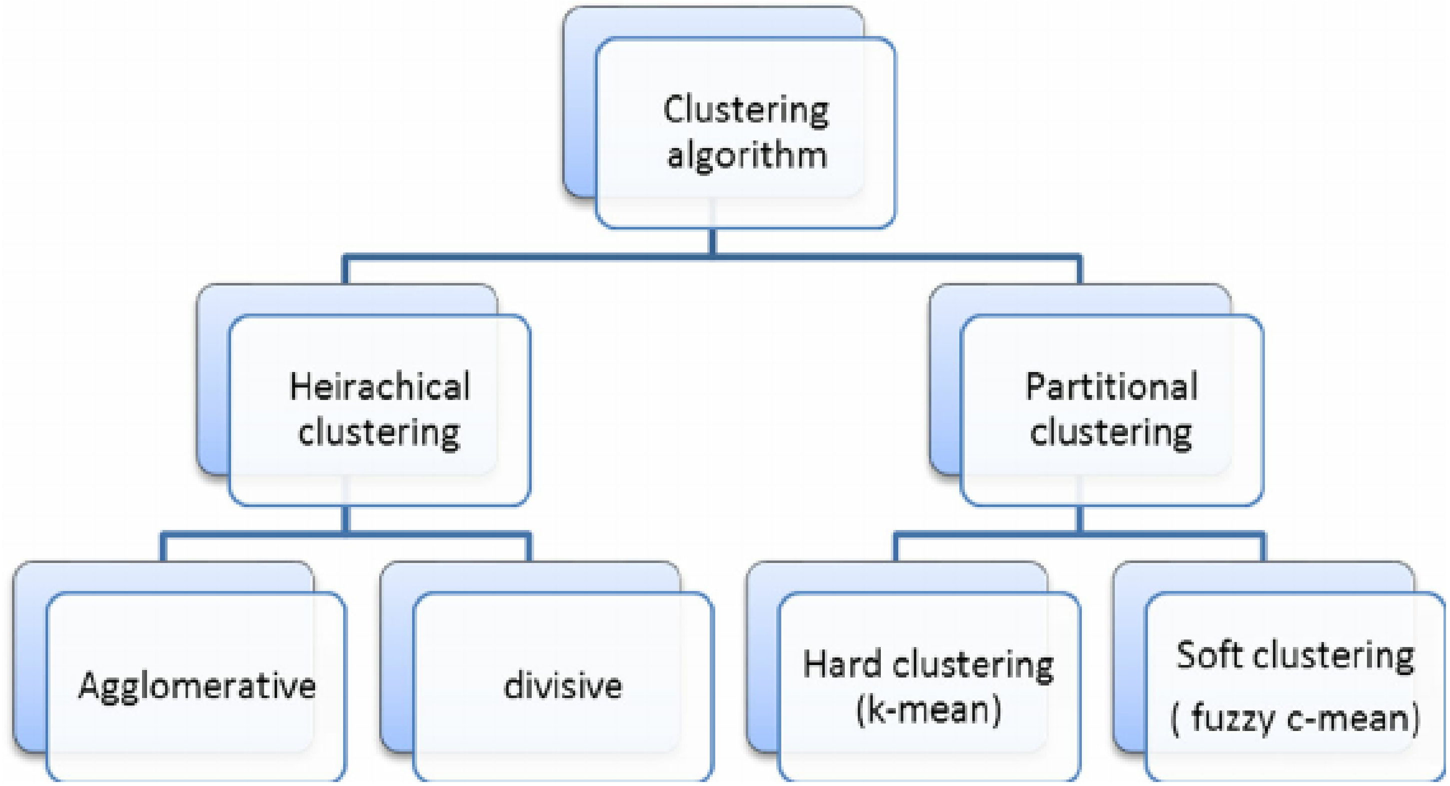


- How would you design an algorithm for finding the three clusters in this case?

Applications of clustering



Classification of clustering algorithm



K-Means clustering

- The k-means algorithm is an algorithm to cluster n objects based on attributes into k partitions, where $k < n$.
- It is similar to the expectation-maximization algorithm for mixtures of Gaussians in that they both attempt to find the centres of natural clusters in the data.
- It assumes that the object attributes form a vector space.

K-Means Clustering

An algorithm for partitioning (or clustering) N data points into K disjoint subsets S_j containing data points so as to minimize the sum-of-squares criterion

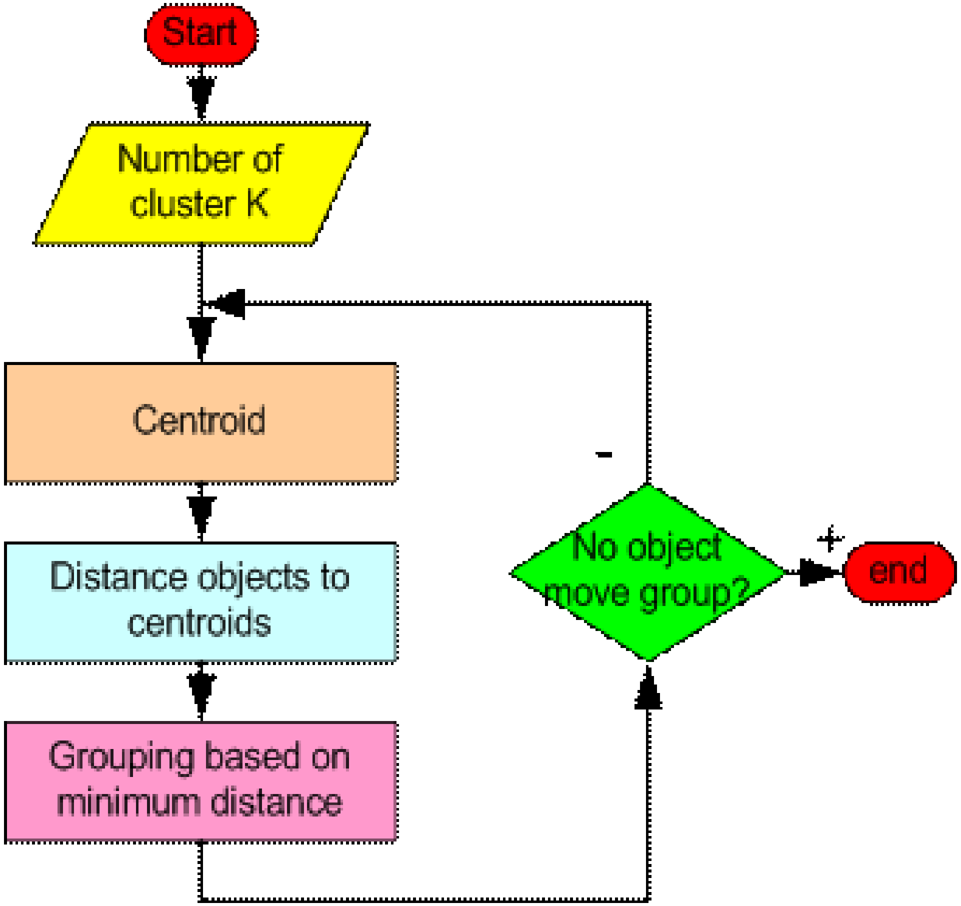
$$J = \sum_{j=1}^{K} \sum_{n \in S_j} \left| x_n - \mu_j \right|^2,$$

where x_n is a vector representing the the n^{th} data point and u_j is the geometric centroid of the data points in S_j.

K-Means clustering

- Simply speaking k-means clustering is an algorithm to classify or to group the objects based on attributes/features into K number of group.
- K is positive integer number.
- The grouping is done by minimizing the sum of squares of distances between data and the corresponding cluster centroid.

How the K-Mean Clustering algorithm works?



K-Mean Algorithm

- **Step 1:** Begin with a decision on the value of k = number of clusters .
- **Step 2**: Put any initial partition that classifies the data into k clusters. You may assign the training samples randomly, or systematically as the following:

  1.Take the first k training sample as single- element clusters

  2. Assign each of the remaining (N-k) training sample to the cluster with the nearest centroid. After each assignment, recompute the centroid of the gaining cluster.

K-Mean Algorithm(Continue…)

- **Step 3:** Take each sample in sequence and compute its distance from the centroid of each of the clusters. If a sample is not currently in the cluster with the closest centroid, switch this sample to that cluster and update the centroid of the cluster gaining the new sample and the cluster losing the sample.
- **Step 4 .** Repeat step 3 until convergence is achieved, that is until a pass through the training sample causes no new assignments.

A Simple example showing the implementation of k-means algorithm
(using K=2)

Individual	Variable 1	Variable 2
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

Step 1:

Initialization: Randomly we choose following two centroids (k=2) for two clusters.

In this case the 2 centroid are: m1=(1.0,1.0) and m2=(5.0,7.0).

Individual	Variable 1	Variable 2
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

	Individual	Mean Vector
Group 1	1	(1.0, 1.0)
Group 2	4	(5.0, 7.0)

Step 2:

- Thus, we obtain two clusters containing:

  {1,2,3} and {4,5,6,7}.

- Their new centroids are:

$$m_1 = \left(\frac{1}{3}(1.0+1.5+3.0), \frac{1}{3}(1.0+2.0+4.0)\right) = (1.83, 2.33)$$

$$m_2 = \left(\frac{1}{4}(5.0+3.5+4.5+3.5), \frac{1}{4}(7.0+5.0+5.0+4.5)\right)$$

$$= (4.12, 5.38)$$

Individual	Centroid 1	Centroid 2
1	0	7.21
2 (1.5, 2.0)	1.12	6.10
3	3.61	3.61
4	7.21	0
5	4.72	2.5
6	5.31	2.06
7	4.30	2.92

$$d(m_1, 2) = \sqrt{|1.0-1.5|^2 + |1.0-2.0|^2} = 1.12$$

$$d(m_2, 2) = \sqrt{|5.0-1.5|^2 + |7.0-2.0|^2} = 6.10$$

Step 3:

- Now using these centroids we compute the Euclidean distance of each object, as shown in table.

- Therefore, the new clusters are:

  {1,2} and {**3**,4,5,6,7}

- Next centroids are: m1=(1.25,1.5) and m2 = (3.9,5.1)

Individual	Centroid 1	Centroid 2
1	1.57	5.38
2	0.47	4.28
3	2.04	1.78
4	5.64	1.84
5	3.15	0.73
6	3.78	0.54
7	2.74	1.08

- Step 4 :

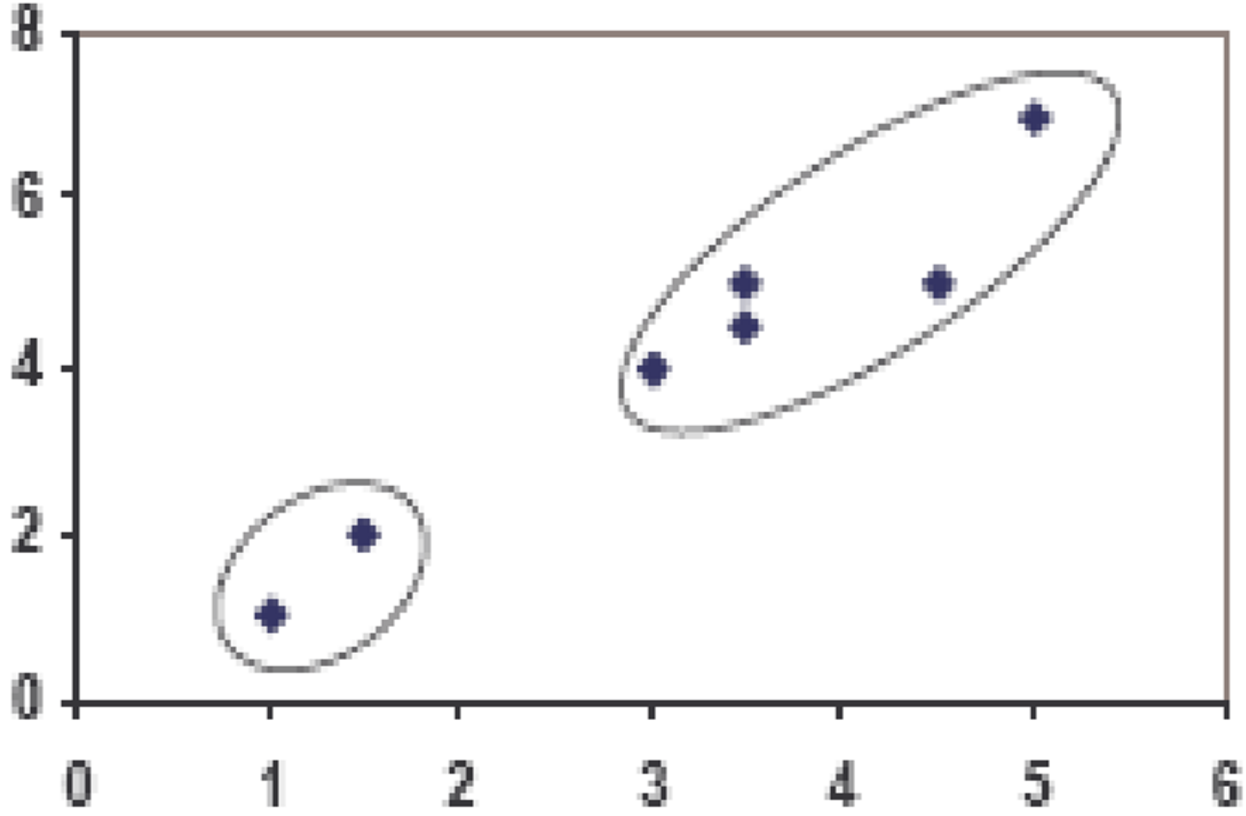
  The clusters obtained are:

  {1,2} and {3,4,5,6,7}

- Therefore, there is no change in the cluster.
- Thus, the algorithm comes to a halt here and final result consist of 2 clusters {1,2} and {3,4,5,6,7}.

Individual	Centroid 1	Centroid 2
1	0.56	5.02
2	0. 56	3.92
3	3.05	1.42
4	6.66	2.20
5	4.16	0.41
6	4.78	0.61
7	3.75	0.72

PLOT



How Many Clusters?

- Number of clusters *K* is given
  - Partition *n* docs into predetermined number of clusters
- Finding the “right” number of clusters is part of the problem
  - Given docs, partition into an “appropriate” number of subsets.
  - E.g., for query results - ideal value of *K* not known up front - though UI may impose limits.
- Can usually take an algorithm for one flavor and convert to the other.

Fuzzy(C-Means) Clustering

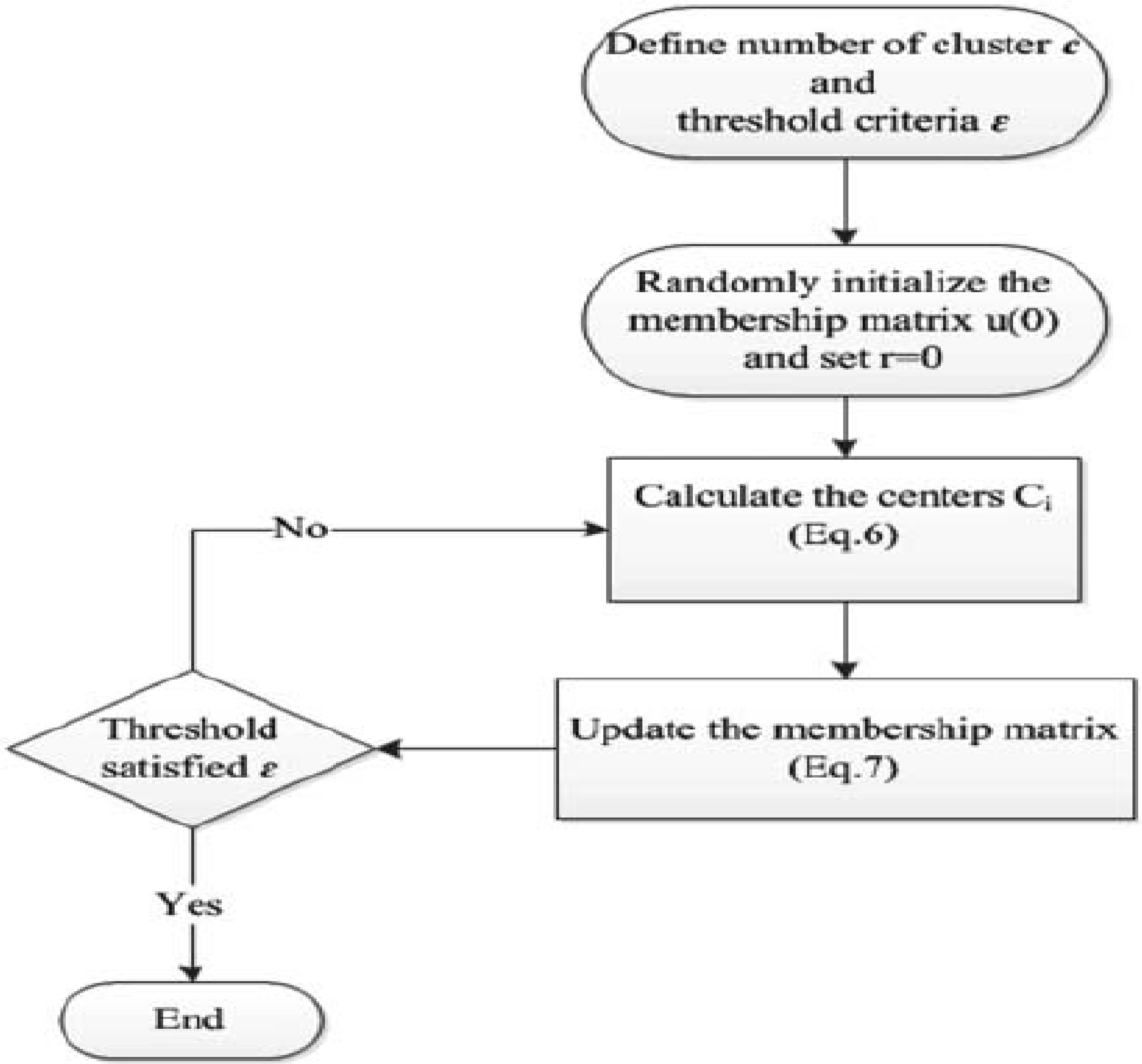
- The unsupervised k-means clustering algorithm gives the values of any point lying in some particular cluster to be either as 0 or 1 i.e., either true or false.
- But the fuzzy logic gives the fuzzy values of any particular data point to be lying in either of the clusters.
- Here, in fuzzy c-means clustering, we find out the centroid of the data points and then calculate the distance of each data point from the given centroids until the clusters formed become constant.

Fuzzy(C-Means) Clustering

- Suppose the given data points are {(1, 3), (2, 5), (6, 8), (7, 9)}

- Fuzzy Clustering is a type of clustering algorithm in machine learning that allows a data point to belong to more than one cluster with different degrees of membership.
- Unlike traditional clustering algorithms, such as k-means or hierarchical clustering, which assign each data point to a single cluster, fuzzy clustering assigns a membership degree between 0 and 1 for each data point for each cluster.

Steps in C-Means Clustering



Applications in several fields of Fuzzy clustering :

- Image segmentation: Fuzzy clustering can be used to segment images by grouping pixels with similar properties together, such as color or texture.
- Pattern recognition: Fuzzy clustering can be used to identify patterns in large datasets by grouping similar data points together.
- Marketing: Fuzzy clustering can be used to segment customers based on their preferences and purchasing behavior, allowing for more targeted marketing campaigns.
- Medical diagnosis: Fuzzy clustering can be used to diagnose diseases by grouping patients with similar symptoms together.
- Environmental monitoring: Fuzzy clustering can be used to identify areas of environmental concern by grouping together areas with similar pollution levels or other environmental indicators.

Advantages of Fuzzy Clustering:

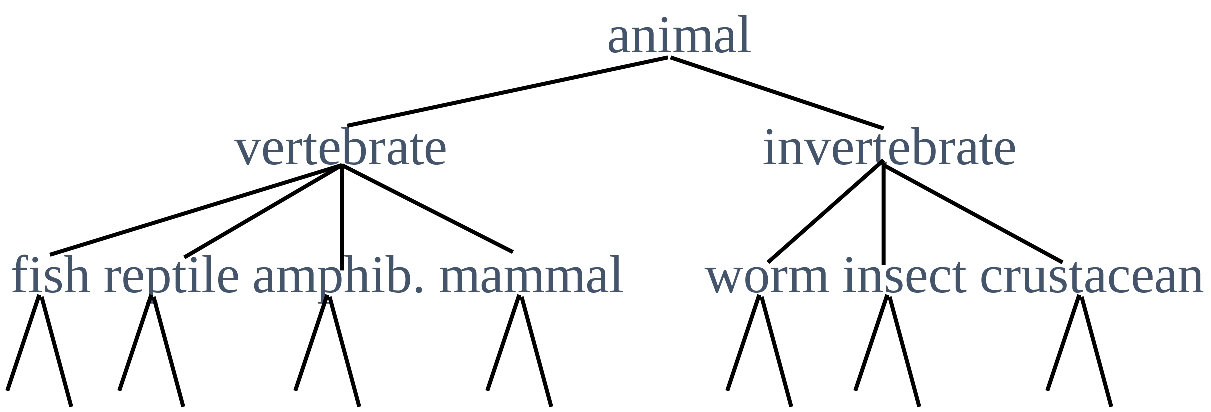
- Flexibility: Fuzzy clustering allows for overlapping clusters, which can be useful when the data has a complex structure or when there are ambiguous or overlapping class boundaries.
- Robustness: Fuzzy clustering can be more robust to outliers and noise in the data, as it allows for a more gradual transition from one cluster to another.
- Interpretability: Fuzzy clustering provides a more nuanced understanding of the structure of the data, as it allows for a more detailed representation of the relationships between data points and clusters.

Disadvantages of Fuzzy Clustering:

- Complexity: Fuzzy clustering algorithms can be computationally more expensive than traditional clustering algorithms, as they require optimization over multiple membership degrees.
- Model selection: Choosing the right number of clusters and membership functions can be challenging, and may require expert knowledge or trial and error.
- If you’re interested in learning more about fuzzy clustering, you might consider reading “Fuzzy Clustering and Its Applications” by James C. Bezdek or “An Introduction to Fuzzy Clustering” by Witold Pedrycz and Fernando Gomide.

Hierarchical Clustering

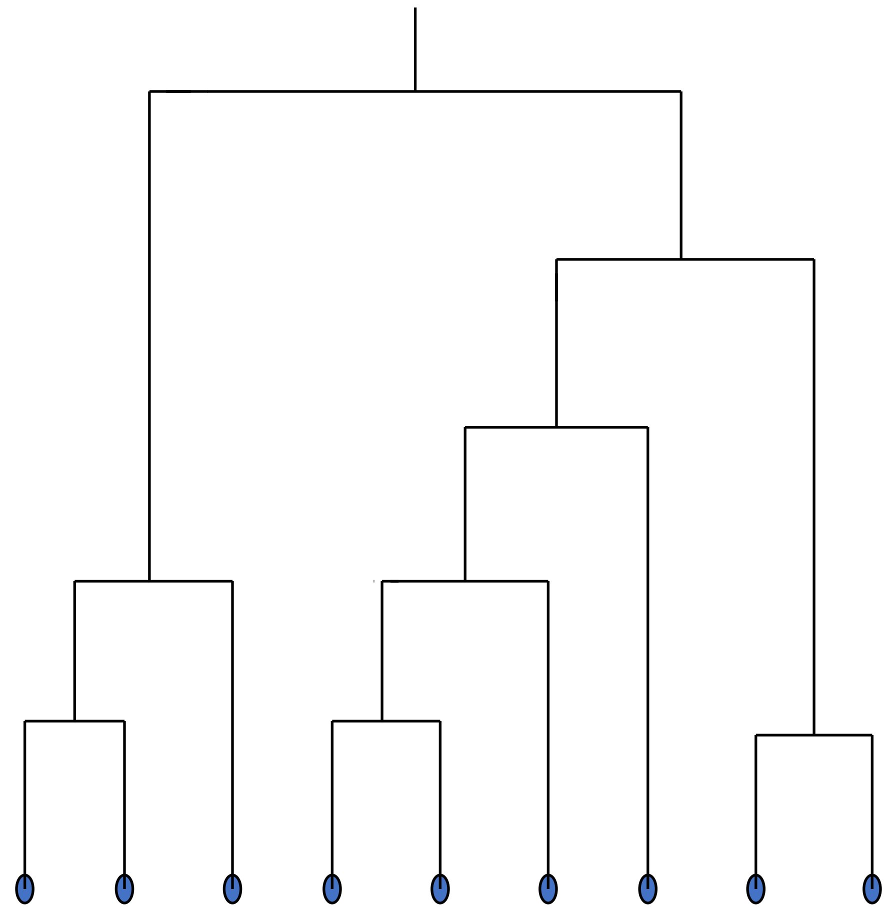
- Build a tree-based hierarchical taxonomy (*dendrogram*) from a set of documents.



- One approach: recursive application of a partitional clustering algorithm.

Dendrogram: Hierarchical Clustering

- Clustering obtained by cutting the dendrogram at a desired level: each connected component forms a cluster.



Types of Hierarchical Clustering

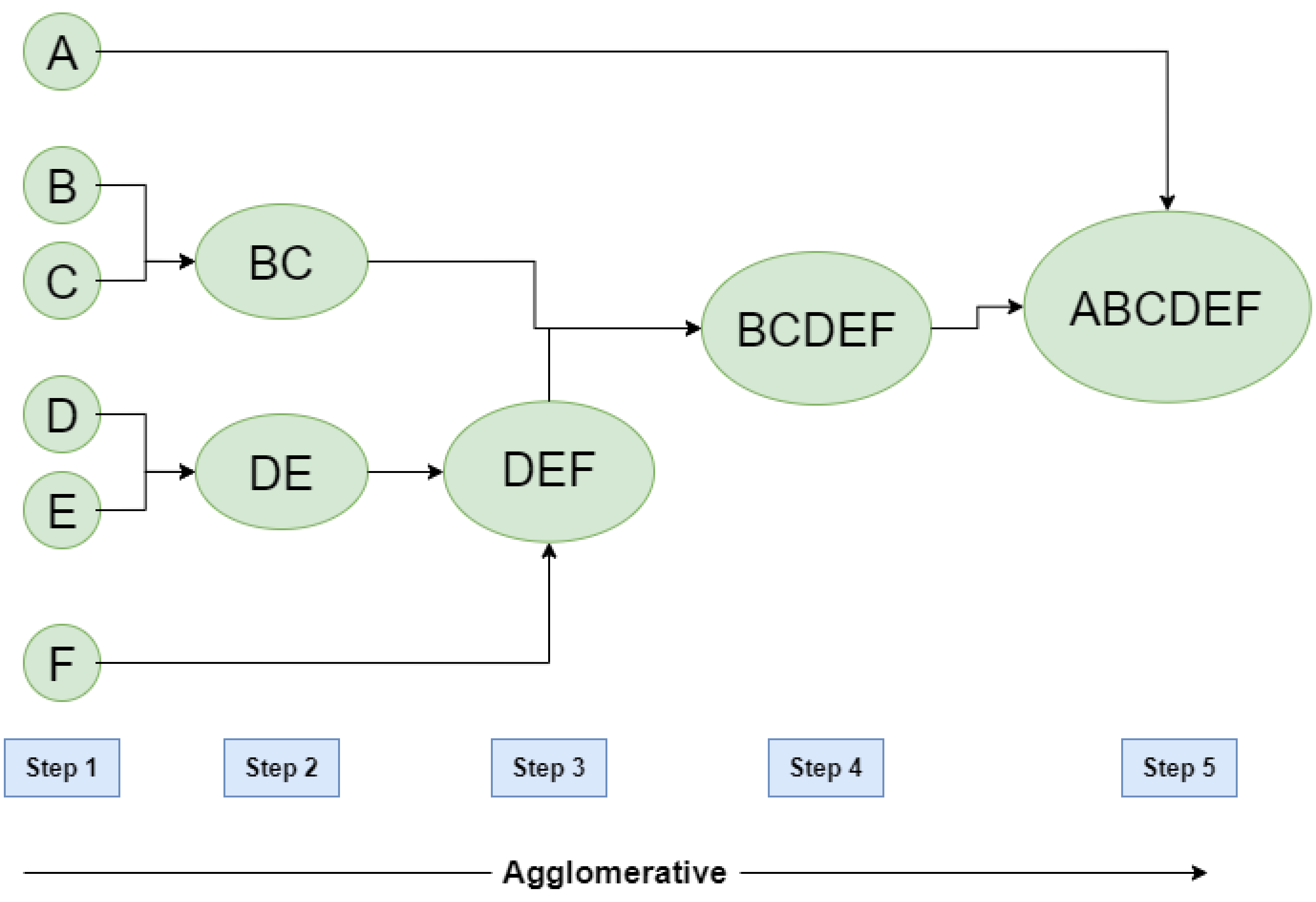
Basically, there are two types of hierarchical Clustering:

- Agglomerative Clustering
- Divisive clustering

Hierarchical Agglomerative Clustering

- It is also known as the bottom-up approach or hierarchical agglomerative clustering (HAC).
- A structure that is more informative than the unstructured set of clusters returned by flat clustering.
- This clustering algorithm does not require us to prespecify the number of clusters.
- Bottom-up algorithms treat each data as a singleton cluster at the outset and then successively agglomerate pairs of clusters until all clusters have been merged into a single cluster that contains all data.

Hierarchical Agglomerative Clustering



Steps in HAC

- Consider each alphabet as a single cluster and calculate the distance of one cluster from all the other clusters.
- In the second step, comparable clusters are merged together to form a single cluster. Let’s say cluster (B) and cluster (C) are very similar to each other therefore we merge them in the second step similarly to cluster (D) and (E) and at last, we get the clusters [(A), (BC), (DE), (F)]
- We recalculate the proximity according to the algorithm and merge the two nearest clusters([(DE), (F)]) together to form new clusters as [(A), (BC), (DEF)]
- Repeating the same process; The clusters DEF and BC are comparable and merged together to form a new cluster. We’re now left with clusters [(A), (BCDEF)].
- At last, the two remaining clusters are merged together to form a single cluster [(ABCDEF)].

Advantages of HAC

- No need for information about how many numbers of clusters are required.
- Easy to use and implement
- We can obtain the optimal number of clusters from the model itself, human intervention not required.
- Dendrograms help us in clear visualization, which is practical and easy to understand.

Disadvantages of HAC

- We can not take a step back in this algorithm.
- Not suitable for large datasets due to high time and space complexity.
- Time complexity is higher at least 0(n^2logn)

“Thank You”

Introduction to Deep Learning Methods and Applications

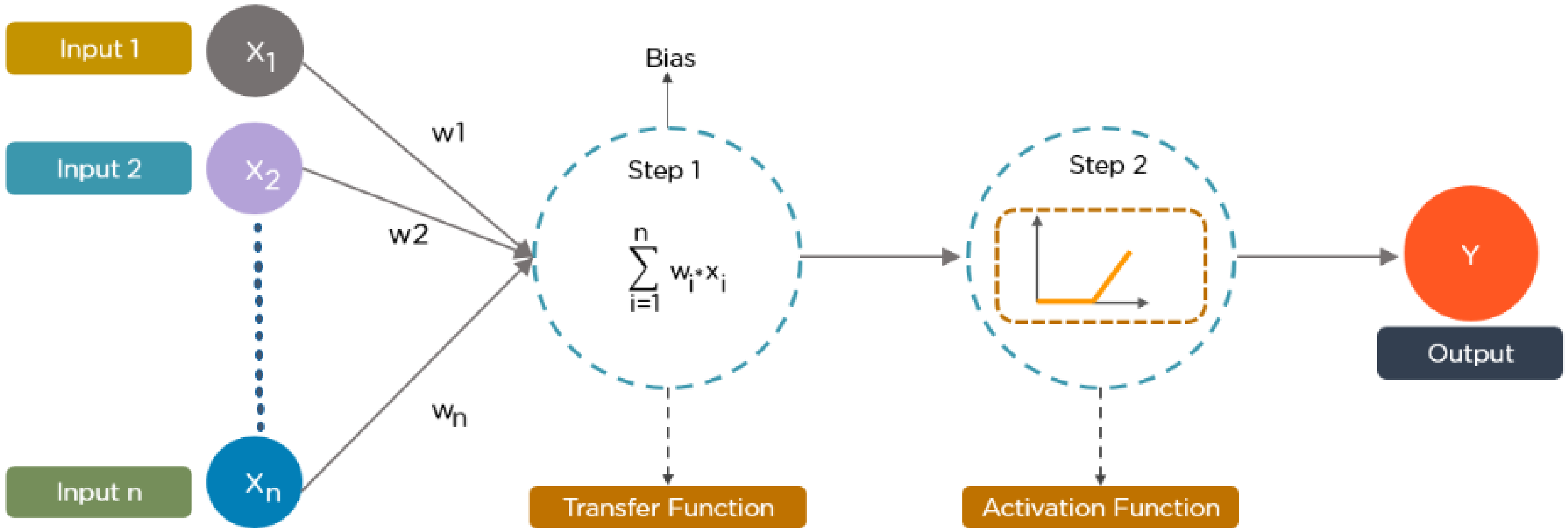
What is Deep Learning?

- Deep learning is based on the branch of machine learning, which is a subset of artificial intelligence.
- It uses artificial neural networks to perform sophisticated computations on large amounts of data. It is a type of machine learning that works based on the structure and function of the human brain.
- Deep learning algorithms train machines by learning from examples. Industries such as health care, eCommerce, entertainment, and advertising commonly use deep learning..
- Deep learning is implemented with the help of Neural Networks, and the idea behind the motivation of Neural Network is the biological neurons, which is nothing but a brain cell.

Defining Neural Networks

A neural network is structured like the human brain and consists of artificial neurons, also known as nodes. These nodes are stacked next to each other in three layers:

- The input layer
- The hidden layer(s)
- The output layer



Data provides each node with information in the form of inputs. The node multiplies the inputs with random weights, calculates them, and adds a bias. Finally, nonlinear functions, also known as activation functions, are applied to determine which neuron to fire.

Deep Learning Algorithms

- Deep learning algorithms work with almost any kind of data and require large amounts of computing power and information to solve complicated issues. Now, let us, deep-dive, into the most popular deep learning algorithms.
- 1. Convolutional Neural Networks (CNNs)
- CNN's, also known as ConvNets, consist of multiple layers and are mainly used for image processing and object detection. Yann LeCun developed the first CNN in 1988 when it was called LeNet. It was used for recognizing characters like ZIP codes and digits.
- CNN's are widely used to identify satellite images, process medical images, forecast time series, and detect anomalies.

How Does CNNs Work?

- CNN's have multiple layers that process and extract features from data:

Convolution Layer

- CNN has a convolution layer that has several filters to perform the convolution operation.

Rectified Linear Unit (ReLU)

- CNN's have a ReLU layer to perform operations on elements. The output is a rectified feature map.

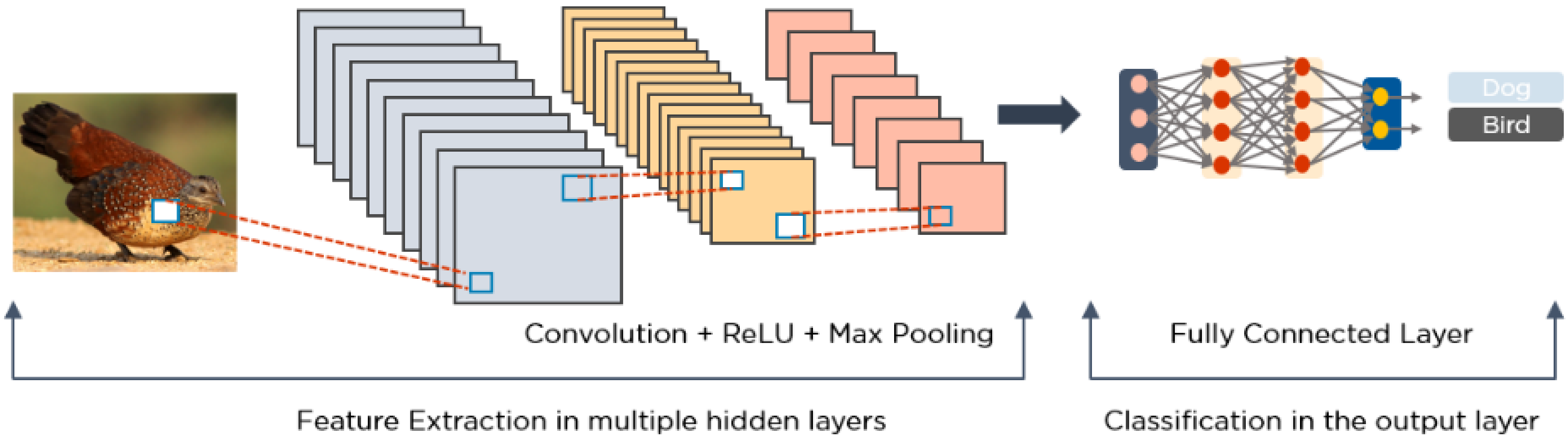
Pooling Layer

- The rectified feature map next feeds into a pooling layer. Pooling is a down-sampling operation that reduces the dimensions of the feature map.
- The pooling layer then converts the resulting two-dimensional arrays from the pooled feature map into a single, long, continuous, linear vector by flattening it.

How Does CNNs Work?(Continue…)

Fully Connected Layer

- A fully connected layer forms when the flattened matrix from the pooling layer is fed as an input, which classifies and identifies the images.
- Below is an example of an image processed via CNN.

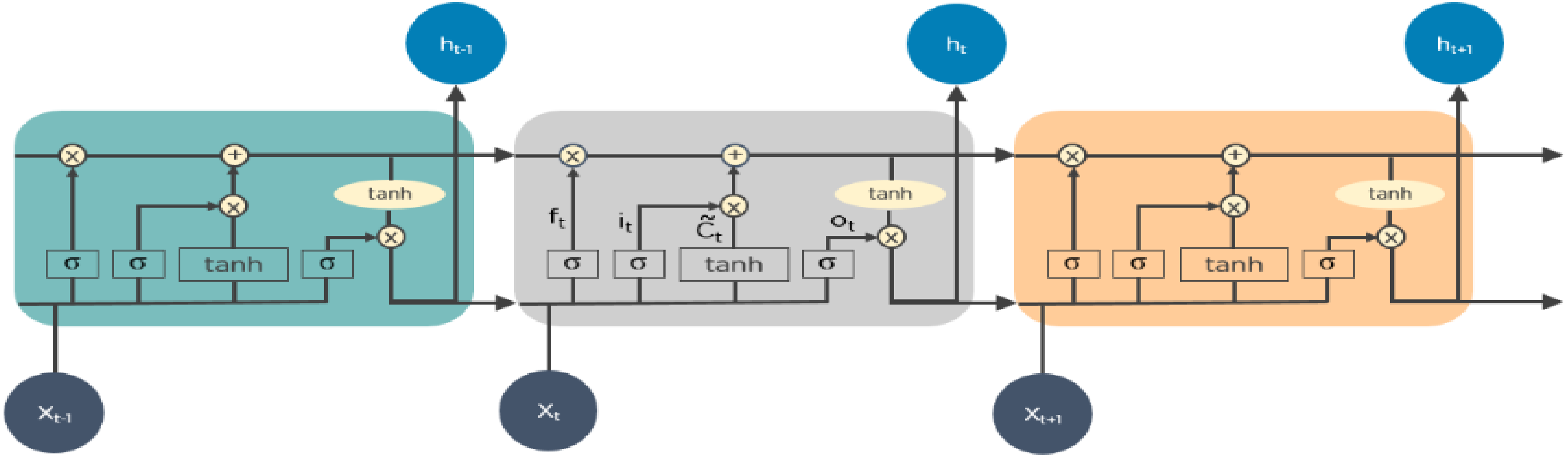


2. Long Short Term Memory Networks (LSTMs)

- LSTMs are a type of Recurrent Neural Network (RNN) that can learn and memorize long-term dependencies. Recalling past information for long periods is the default behaviour.
- LSTMs retain information over time. They are useful in time-series prediction because they remember previous inputs.
- LSTMs have a chain-like structure where four interacting layers communicate in a unique way. Besides time-series predictions, LSTMs are typically used for speech recognition, music composition, and pharmaceutical development.

How Do LSTMs Work?

- First, they forget irrelevant parts of the previous state
- Next, they selectively update the cell-state values
- Finally, the output of certain parts of the cell state
- Below is a diagram of how LSTMs operate:

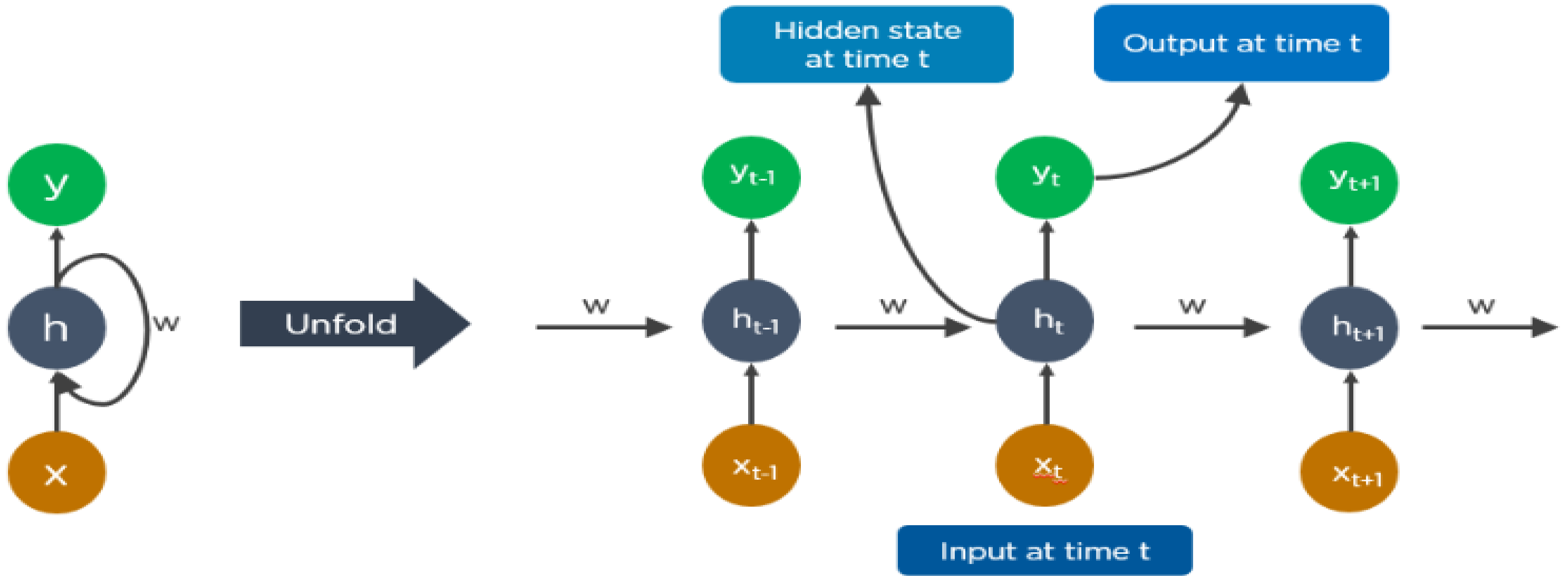


3.Recurrent Neural Networks(RNNs)

- RNNs have connections that form directed cycles, which allow the outputs from the LSTM to be fed as inputs to the current phase.
- The output from the LSTM becomes an input to the current phase and can memorize previous inputs due to its internal memory.
- RNNs are commonly used for image captioning, time-series analysis, natural-language processing, handwriting recognition, and machine translation.

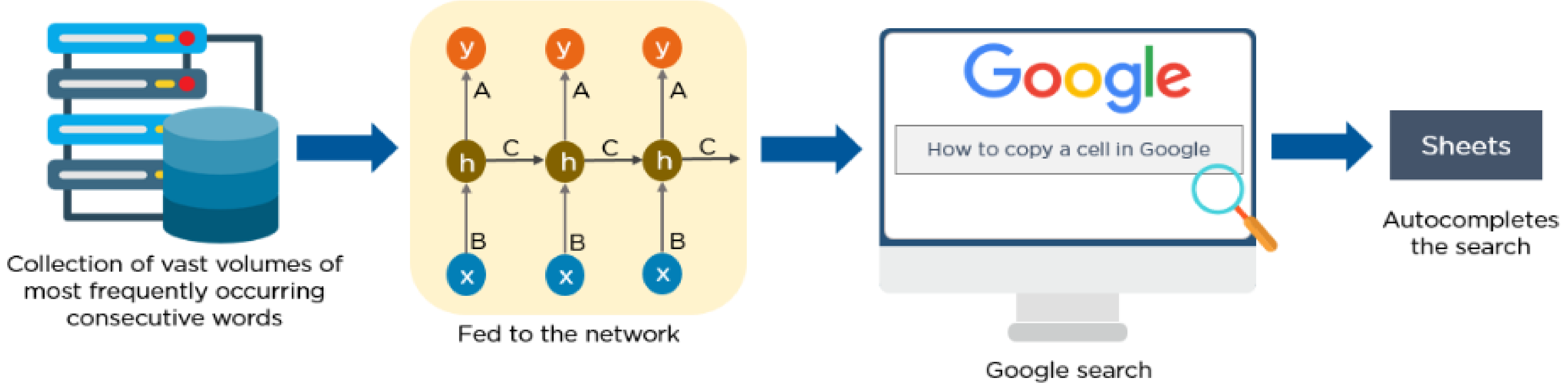
3.Recurrent Neural Networks(RNNs)(Continue..)

An unfolded RNN looks like this:



How Does RNNs work?

- The output at time t-1 feeds into the input at time t.
- Similarly, the output at time t feeds into the input at time t+1.
- RNNs can process inputs of any length.
- The computation accounts for historical information, and the model size does not increase with the input size.
- Here is an example of how Google's autocompleting feature works:



4. Generative Adversarial Networks (GANs)

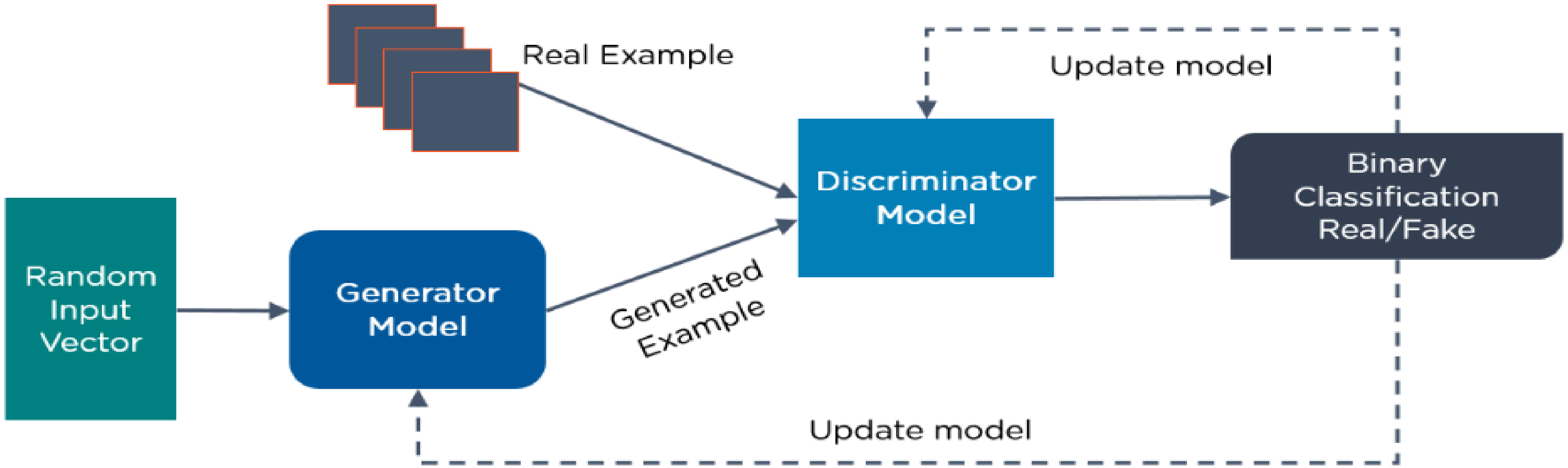
- GANs are generative deep learning algorithms that create new data instances that resemble the training data. GAN has two components: a generator, which learns to generate fake data, and a discriminator, which learns from that false information.

- The usage of GANs has increased over a period of time. They can be used to improve astronomical images and simulate gravitational lensing for dark-matter research. Video game developers use GANs to upscale low-resolution, 2D textures in old video games by recreating them in 4K or higher resolutions via image training.

- GANs help generate realistic images and cartoon characters, create photographs of human faces, and render 3D objects.

How Does GANs work?

- The discriminator learns to distinguish between the generator’s fake data and the real sample data.
- During the initial training, the generator produces fake data, and the discriminator quickly learns to tell that it's false.
- The GAN sends the results to the generator and the discriminator to update the model.
- Below is a diagram of how GANs operate:



5. Deep Belief Networks (DBNs)

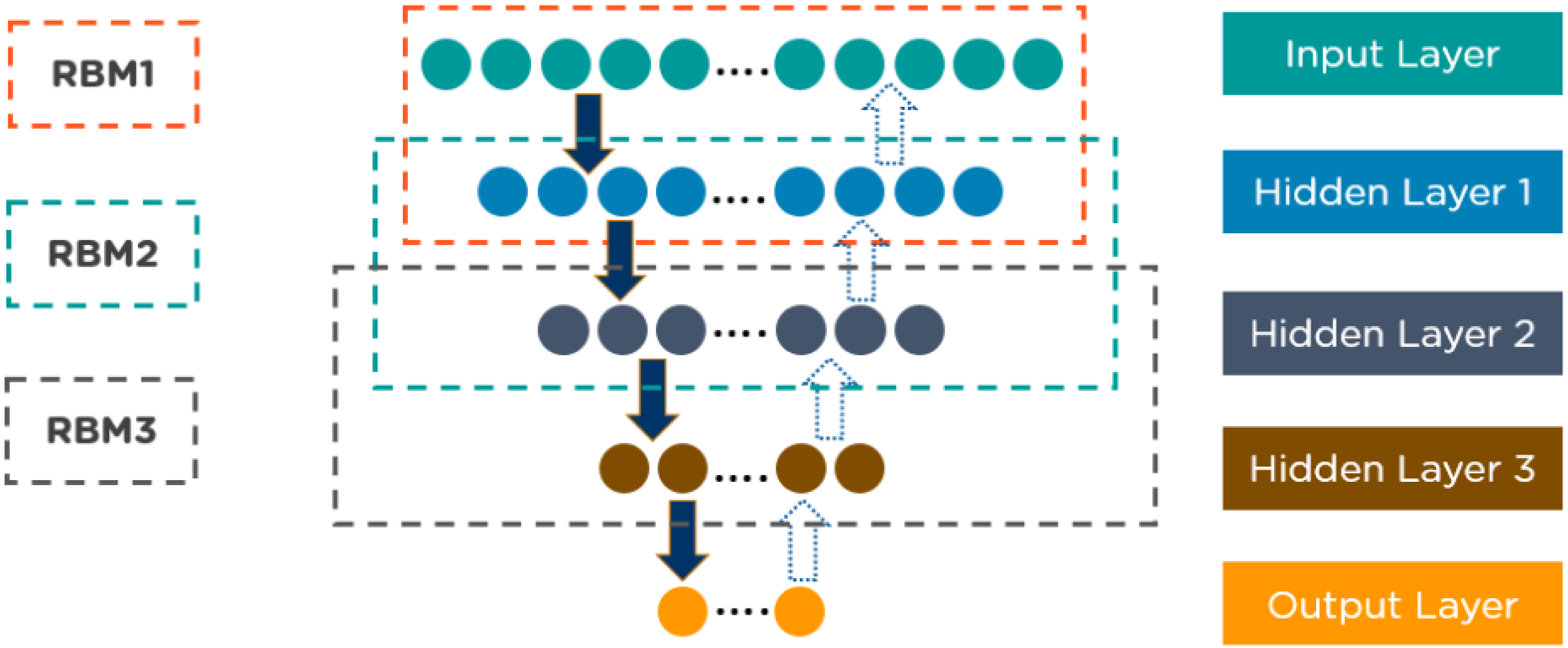
- DBNs are generative models that consist of multiple layers of stochastic, latent variables. The latent variables have binary values and are often called hidden units.
- DBNs are a stack of Boltzmann Machines with connections between the layers, and each RBM layer communicates with both the previous and subsequent layers.
- Deep Belief Networks (DBNs) are used for image-recognition, video-recognition, and motion-capture data.

How Do DBNs Work?

- Greedy learning algorithms train DBNs. The greedy learning algorithm uses a layer-by-layer approach for learning the top-down, generative weights.
- DBNs run the steps of Gibbs sampling on the top two hidden layers. This stage draws a sample from the RBM defined by the top two hidden layers.
- DBNs draw a sample from the visible units using a single pass of ancestral sampling through the rest of the model.
- DBNs learn that the values of the latent variables in every layer can be inferred by a single, bottom-up pass.

How Do DBNs Work?(Continue…)

- Below is an example of DBN architecture:



6. Restricted Boltzmann Machines (RBMs)

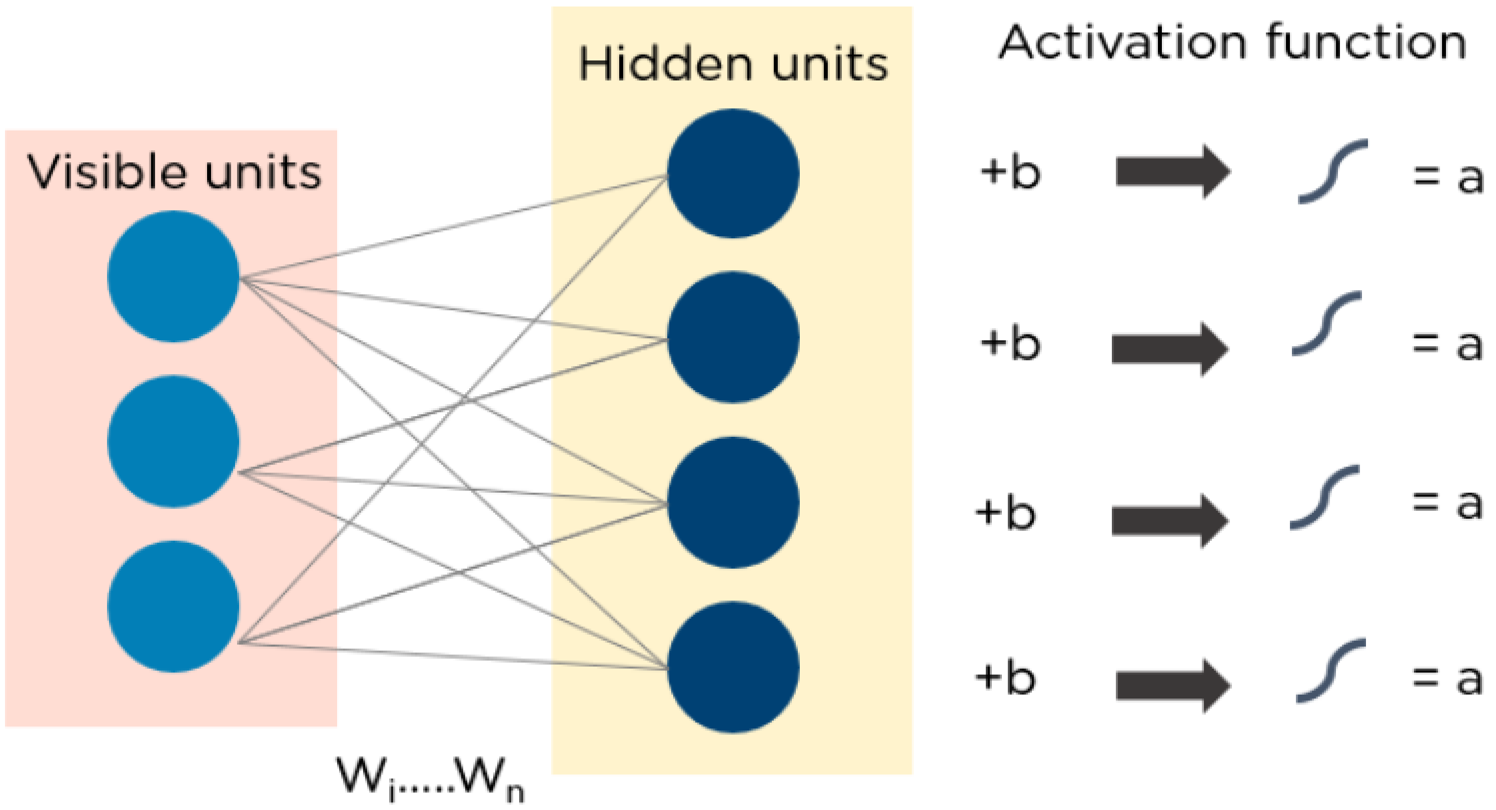
- Developed by Geoffrey Hinton, RBMs are stochastic neural networks that can learn from a probability distribution over a set of inputs.
- This deep learning algorithm is used for dimensionality reduction, classification, regression, collaborative filtering, feature learning, and topic modeling. RBMs constitute the building blocks of DBNs.
- RBMs consist of two layers:
- Visible units
- Hidden units
- Each visible unit is connected to all hidden units. RBMs have a bias unit that is connected to all the visible units and the hidden units, and they have no output nodes.

How Does RBMs Work?

- RBMs have two phases: forward pass and backward pass.
- RBMs accept the inputs and translate them into a set of numbers that encodes the inputs in the forward pass.
- RBMs combine every input with individual weight and one overall bias. The algorithm passes the output to the hidden layer.
- In the backward pass, RBMs take that set of numbers and translate them to form the reconstructed inputs.
- RBMs combine each activation with individual weight and overall bias and pass the output to the visible layer for reconstruction.
- At the visible layer, the RBM compares the reconstruction with the original input to analyze the quality of the result.

How Does RBMs Work?(Continue..)

- Below is a diagram of how RBMs function



7. Autoencoders

- Autoencoders are a specific type of feedforward neural network in which the input and output are identical.
- Geoffrey Hinton designed autoencoders in the 1980s to solve unsupervised learning problems.
- They are trained neural networks that replicate the data from the input layer to the output layer.
- Autoencoders are used for purposes such as pharmaceutical discovery, popularity prediction, and image processing.

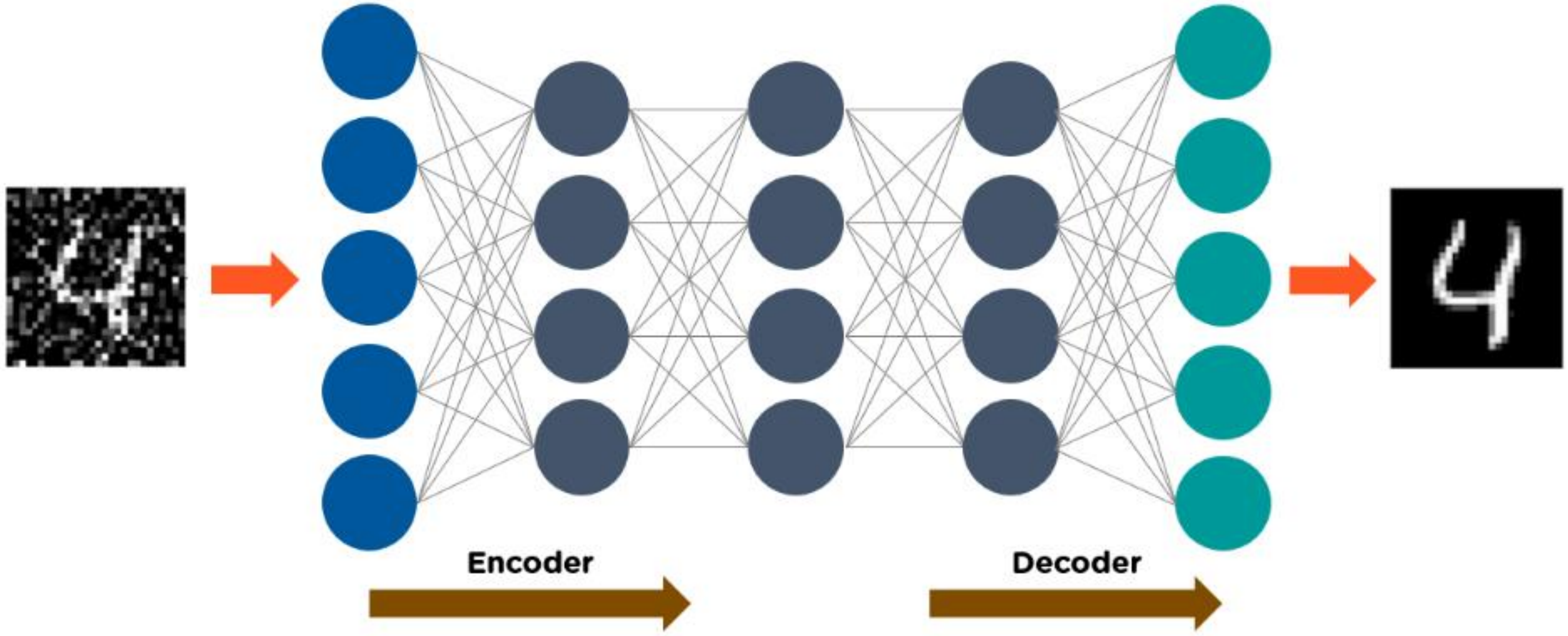
How Do Autoencoders Work?

- An autoencoder consists of three main components: the encoder, the code, and the decoder.

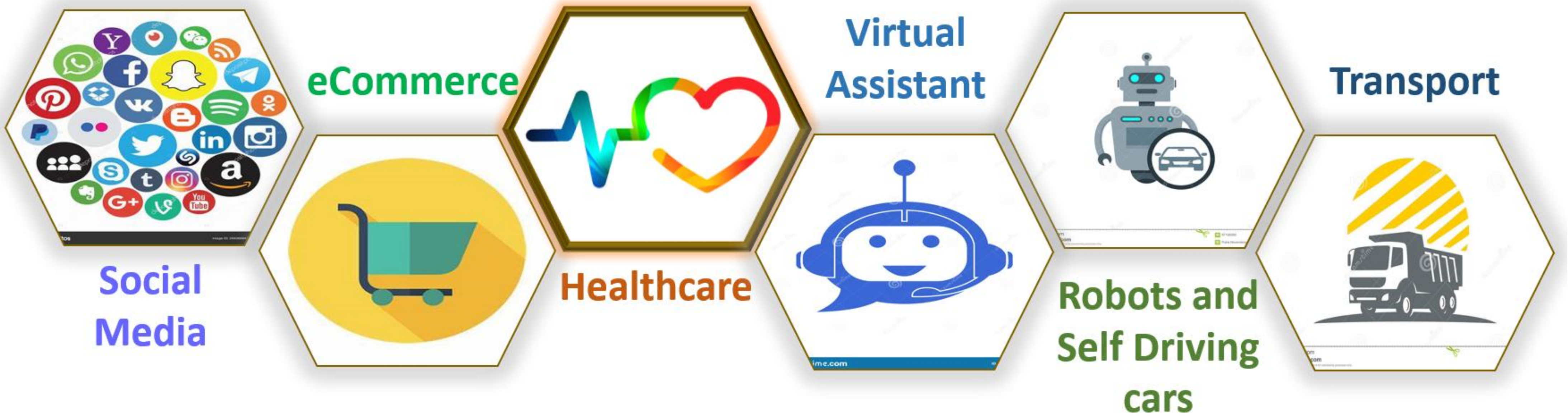
- Autoencoders are structured to receive an input and transform it into a different representation. They then attempt to reconstruct the original input as accurately as possible.
- When an image of a digit is not clearly visible, it feeds to an autoencoder neural network.
- Autoencoders first encode the image, then reduce the size of the input into a smaller representation.
- Finally, the autoencoder decodes the image to generate the reconstructed image.
- The following image demonstrates how autoencoders operate:

How Do Autoencoders Work?(Continue…)

- The following image demonstrates how autoencoders operate:



Applications of Deep Learning

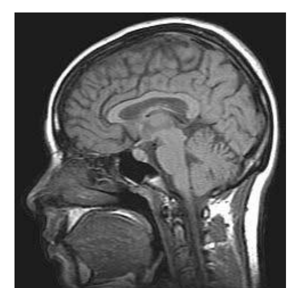


“Thank You”

Machine Learning Applications in Computer Vision

What is Computer Vision?

- **Computer vision** is the science and technology of machines that see.
- Concerned with the theory for building artificial systems that obtain information from images.
- The image data can take many forms, such as a video sequence, depth images, views from multiple cameras, or multi-dimensional data from a medical scanner



Computer Vision

Make computers understand images and videos.

What kind of scene?

Where are the cars?

How far is the building?

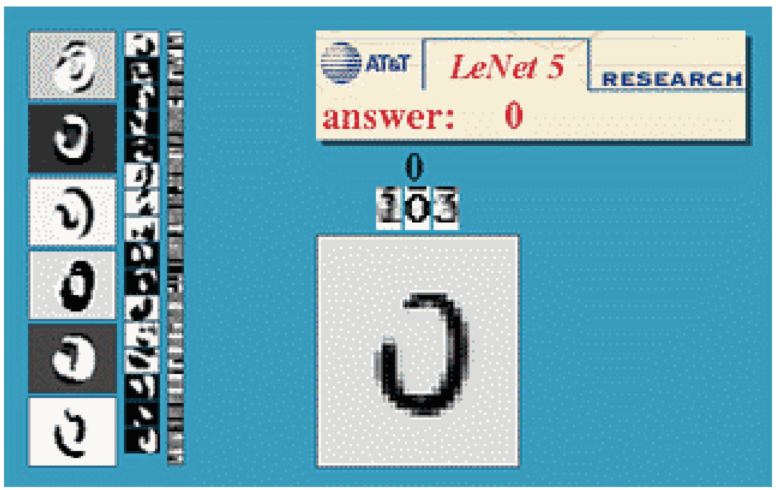
…

Computer vision can help in

Optical character recognition (OCR)

Technology to convert scanned docs to text

- If you have a scanner, it probably came with OCR software



Digit recognition, AT&T labs

/

License plate readers

Face detection

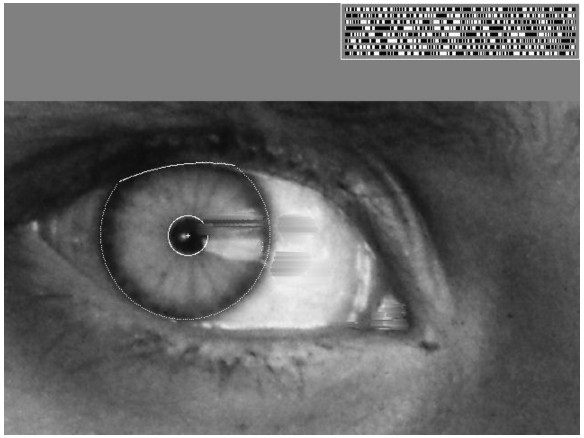
- Many new digital cameras now detect faces
  - Canon, Sony, Fuji, …

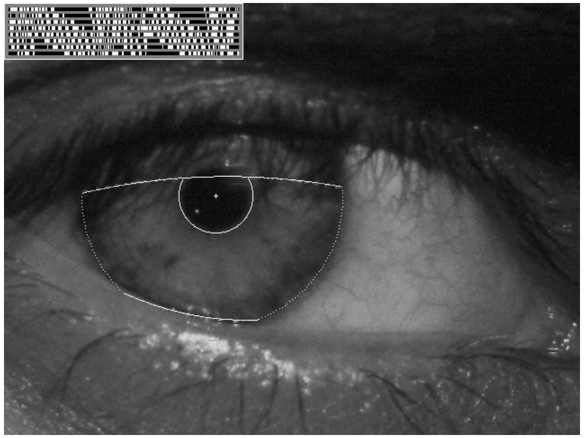
Object recognition (in supermarkets)

"A smart camera is flush-mounted in the checkout lane, continuously watching for items. When an item is detected and recognized, the cashier verifies the quantity of items that were found under the basket, and continues to close the transaction. The item can remain under the basket, and with LaneHawk,you are assured to get paid for it… "

Vision-based biometrics

"*How the Afghan Girl was Identified by Her Iris Patterns*"





What is video surveillance?

Video surveillance involves the act of observing a scene or scenes and looking for specific behaviors that are improper or that may indicate the emergence or existence of improper behavior.

- **Present Implementations?**

  -Human detection systems.

  -Vehicle monitoring systems.

- **Advantages of video surveillance?**

  -Keep track of information video data for future use.

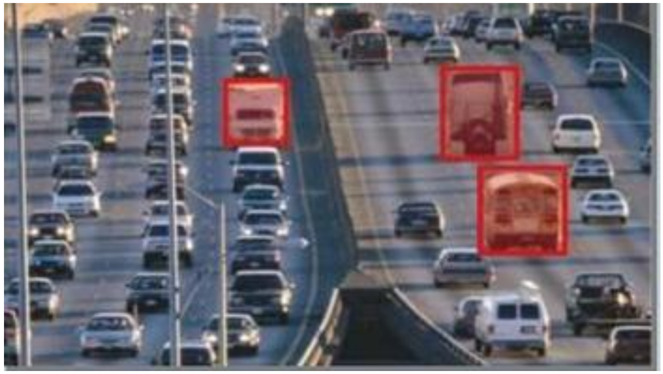
  - Helpful in identifying people in the crime scenes etc..

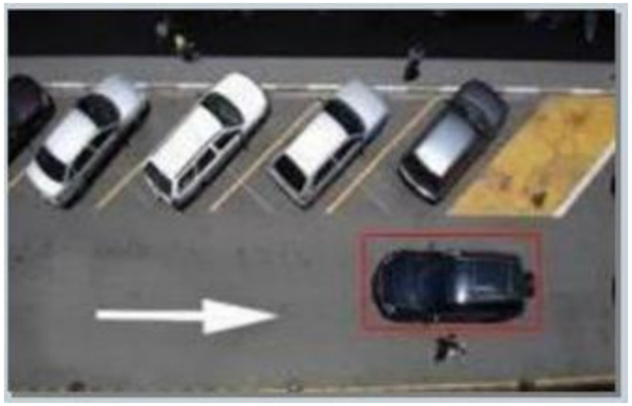
Why Video Surveillance?

- Improve public safety
- Mitigate risks of crime and terrorism
- Protect assets
- Prevent fraud
- Improve efficiency
- Provide better healthcare

Video Surveillance Applications: Forbidden Zone Alarm

- Forbidden Zone protection
- Intruder detection
- Object tracing •
- Object detection of moving trail in open area





Video Surveillance Applications: Behavior Analysis

Personal injury detection

- Public responsibility
- Elderly home care
- Instant assistance and prevention

Suspicious behavior detection

- Loaf and run behavior
- Custom behavior detection model

Video surveillance in the context of Computer Vision

- Detection and tracking of moving objects are the important tasks of the computer vision.
- The video surveillance systems not only need to track the moving objects but also interpret their patterns of behaviours. This means solving the information and integration the pattern.

Advantages

• Minimizes the user interaction.

• Less amount of prohibitive bandwidth. • Minimizes the cost and time.

Need for Traffic Monitoring

- To reduce the traffic congestion on highways
- Reduce the road accidents
- Identifying suspicious vehicles. Etc..,

Traffic Monitoring in Computer Vision

- The quest for better traffic information, an increasing reliance on traffic surveillance has resulted in a better vehicle detection.
- Taking some intelligent actions based on the conditions.

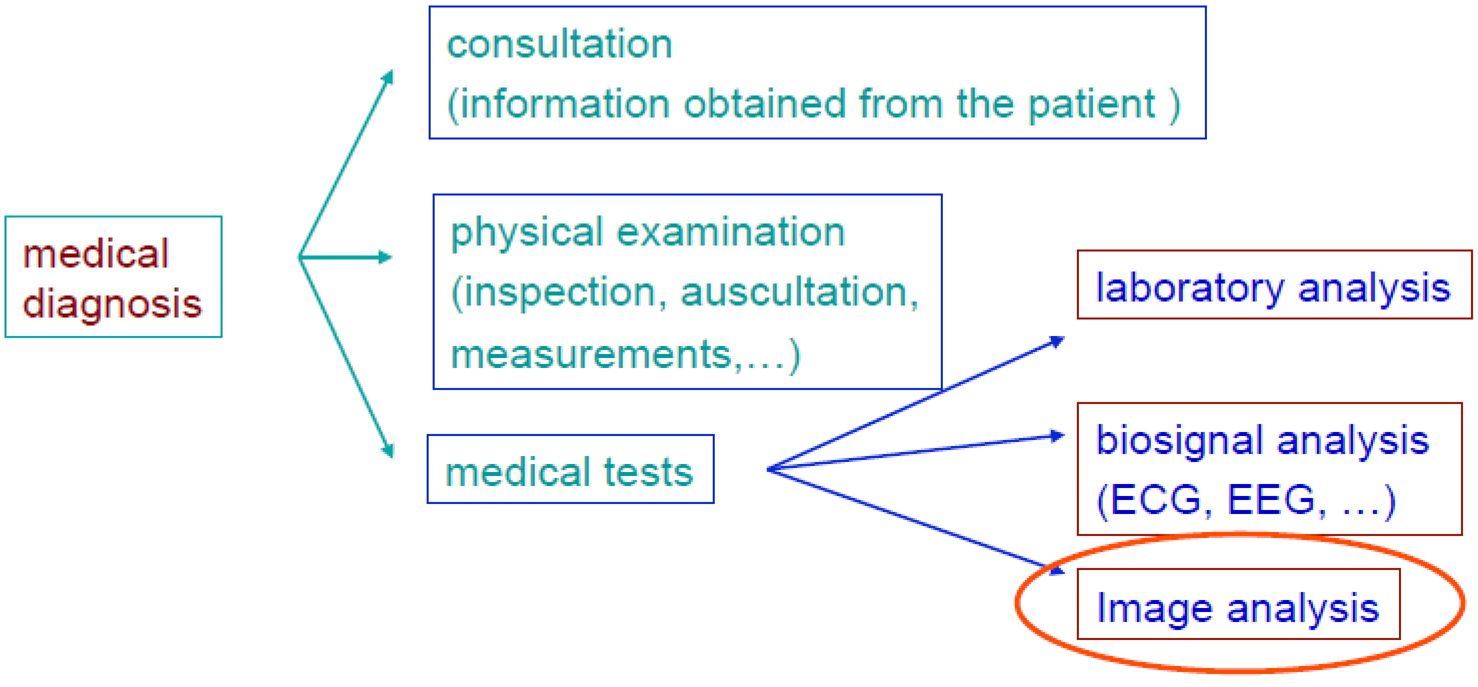
Traffic scene analysis in 3 categories.

- A strait forward vehicle detection and counting system .
- Congestion monitoring and traffic scene analysis.
- Vehicle classification and tracking systems which involve much more detailed scene traffic analysis.

Medical Imaging

- Medical imaging is the technique and process of creating visual representations of the interior of a body for clinical analysis and medical intervention, as well as visual representation of the function of some organs or tissues

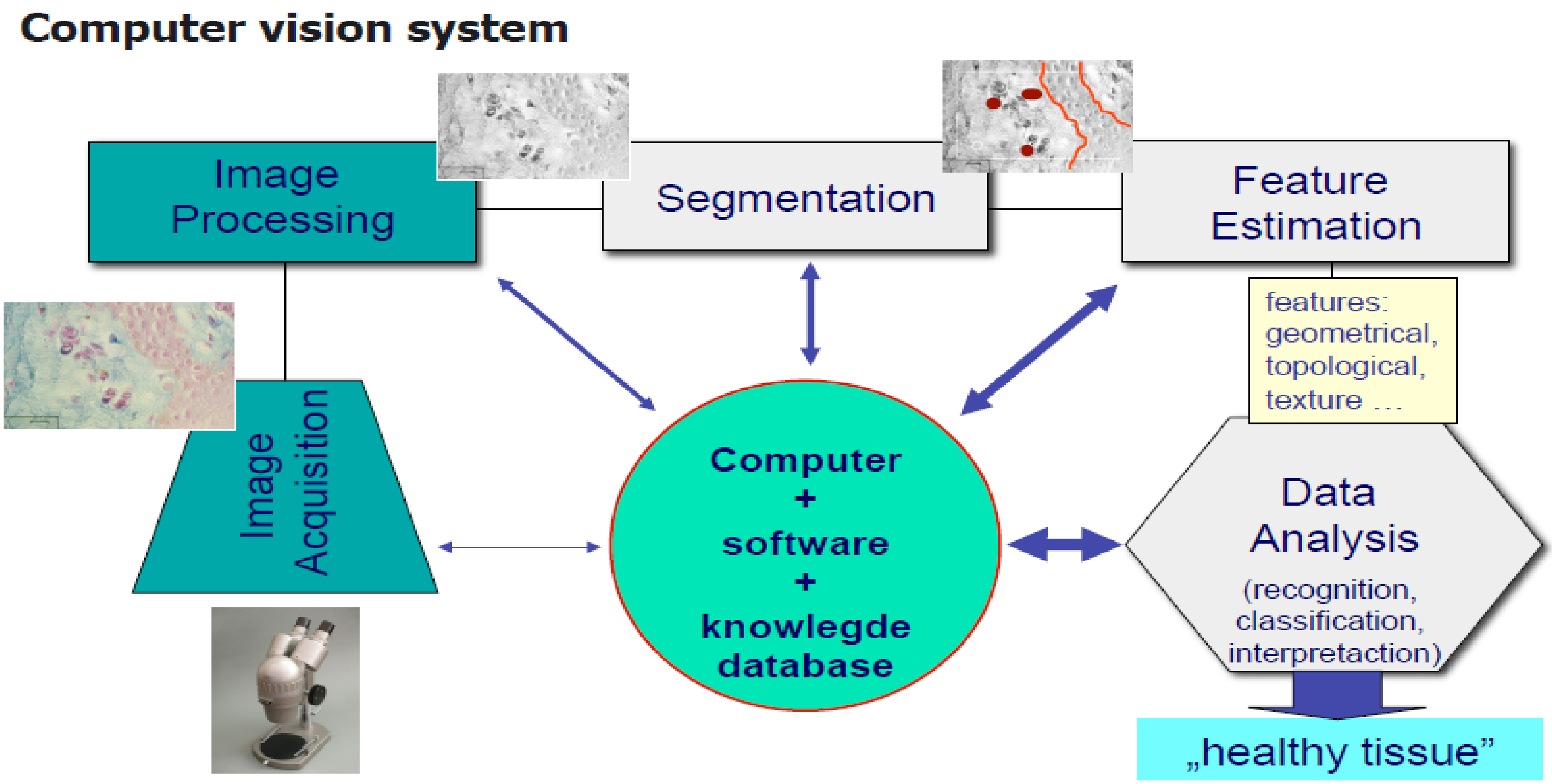
Medical Imaging



Why is Medical Imaging Important?

- To diagnose, treat and cure patients without causing any harmful side effects.
- Computer vision can exploit texture, shape, contour and prior knowledge along with contextual information from image sequence and provide 3D and 4D information that helps with better human understanding.
- To see inside a patient without having to cut them open
- The final objective is to benefit the patients without adding to the already high medical costs.

Computer Vision System



What Are The Different Types Of Medical Imaging?

- Medical imaging is an umbrella to several similar technologies that uses non-invasive imagery to take different parts of the patient’s body via:
- X-ray
- Ultrasound
- Magnetic Response Imagine (MRI)
- Computed Tomography (CT)
- Tactile Imaging
- Endoscopy

Benefits Of Medical Imaging

I has become reliable and accurate to distinguish, recognize, and identify abnormalities in various applications. Medical Imaging integrates AI to identify early traces of diseases where the human eye cannot. By detecting these issues in a timely manner, an appropriate diagnosis can be made, and treatment can begin before it develops to something more serious. Medical Imaging is capable of detecting various abnormalities:

- Tumors
- Cancer
- Stroke
- Pneumonia
- COVID
- … and more

Benefits Of Medical Imaging

Improved Insight Data Analysis For Decision Making

- Doctors and surgeons need as much information as possible to provide an accurate diagnosis or procedure.
- Medical Imaging has aided healthcare professionals with detailed imagery and more reliable information to develop a much more accurate diagnosis.
- Misidentifying an issue can cause irreversible damage and consequences for both the patient and doctor. With more intricate data, the odds of these incidents occurring significantly decrease.

Benefits Of Medical Imaging

Monitoring

- Medical Imaging not only provides early detection, but it is also capable of monitoring abnormalities.
- Patients that are recovering from treatment can be evaluated over time to ensure when they are fully recovered.
- Some patients with non-life-threatening conditions need to be monitored for practitioners to decide if an operation will be needed or instruct the patient to follow a routine that will ease their condition naturally.

“Thank You”

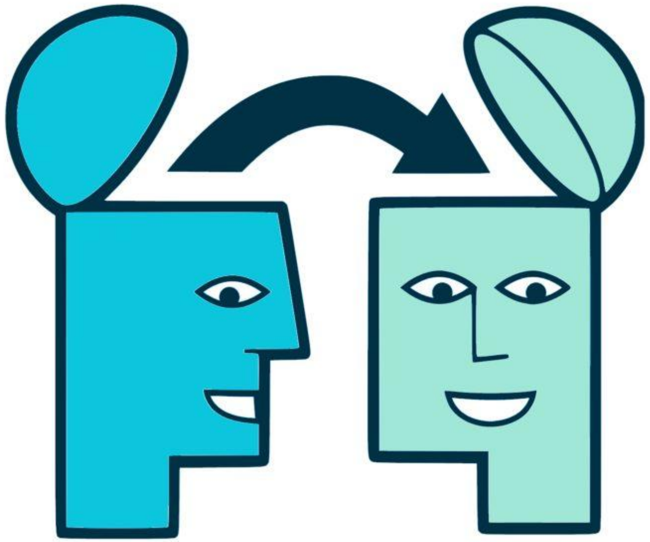
Transfer Learning

Motivation

- Lots of data, time, resources needed to train and tune a neural network from scratch
- An ImageNet deep neural net can take weeks to train and fine-tune from scratch.
- Unless you have 256 GPUs, possible to achieve in 1 hour
- Cheaper, faster way of adapting a neural network by exploiting their generalization properties

What is Transfer Learning?

- Transfering the knowledge of one model to perform a new task.
- Transfer learning is a powerful technique used in Deep Learning.
- By harnessing the ability to reuse existing models and their knowledge of new problems, transfer learning has opened doors to training deep neural networks even with limited data.
- This breakthrough is especially significant in data science, where practical scenarios often need more labeled data.

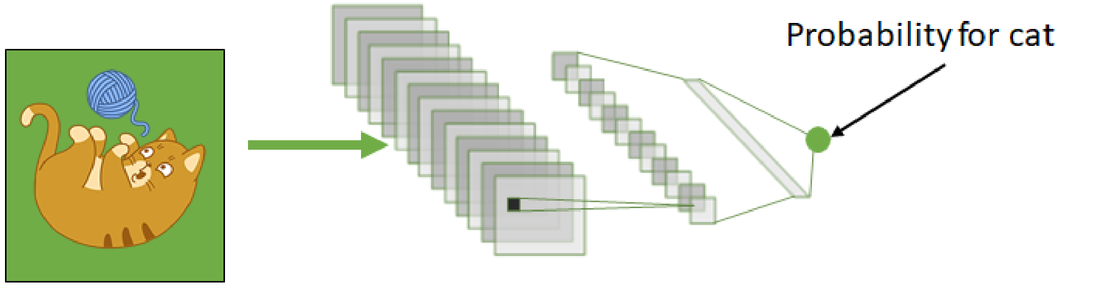


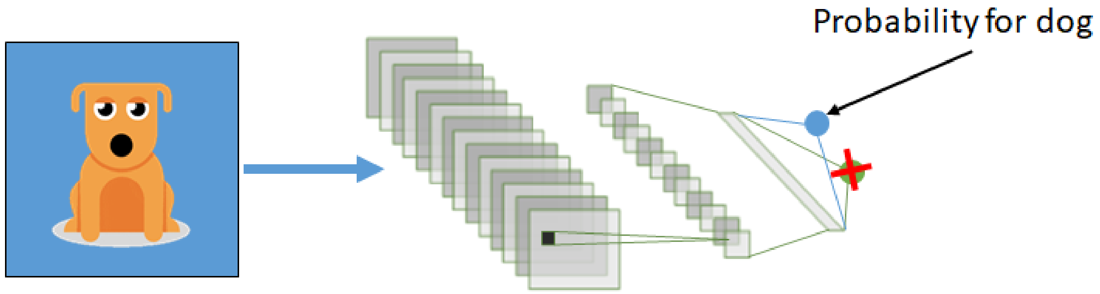
What is Transfer Learning?(Continue..)

- The reuse of a pre-trained model on a new problem is known as transfer learning in machine learning.
- A machine uses the knowledge learned from a prior assignment to increase prediction about a new task in transfer learning. You could, for example, use the information gained during training to distinguish beverages when training a classifier to predict whether an image contains cuisine.
- The knowledge of an already trained machine learning model is transferred to a different but closely linked problem throughout transfer learning.
- For example, if you trained a simple classifier to predict whether an image contains a backpack, you could use the model's training knowledge to identify other objects such as sunglasses.

What is Transfer Learning?(Continue..)

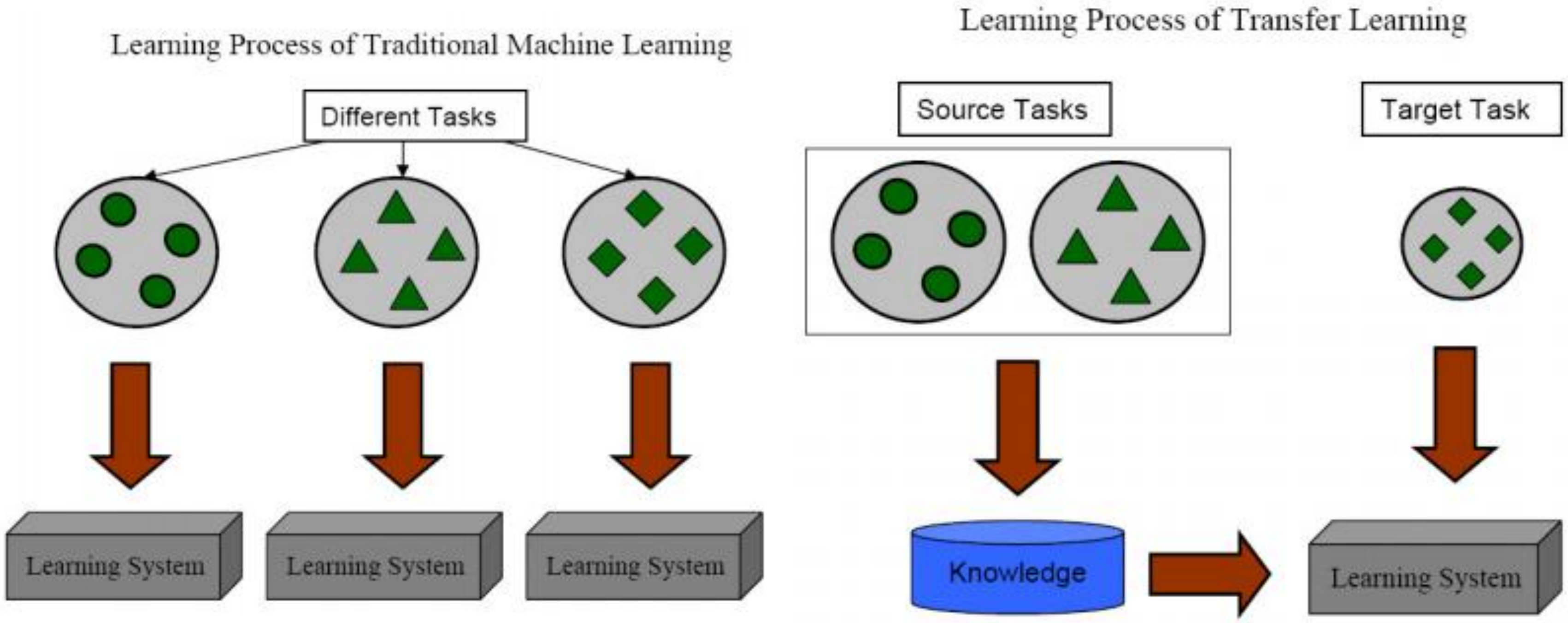
- With transfer learning, we basically try to use what we've learned in one task to better understand the concepts in another. weights are being automatically being shifted to a network performing "task A" from a network that performed new "task B."





- Because of the massive amount of CPU power required, transfer learning is typically applied in computer vision and natural language processing tasks like sentiment analysis.

Traditional vs. Transfer Learning

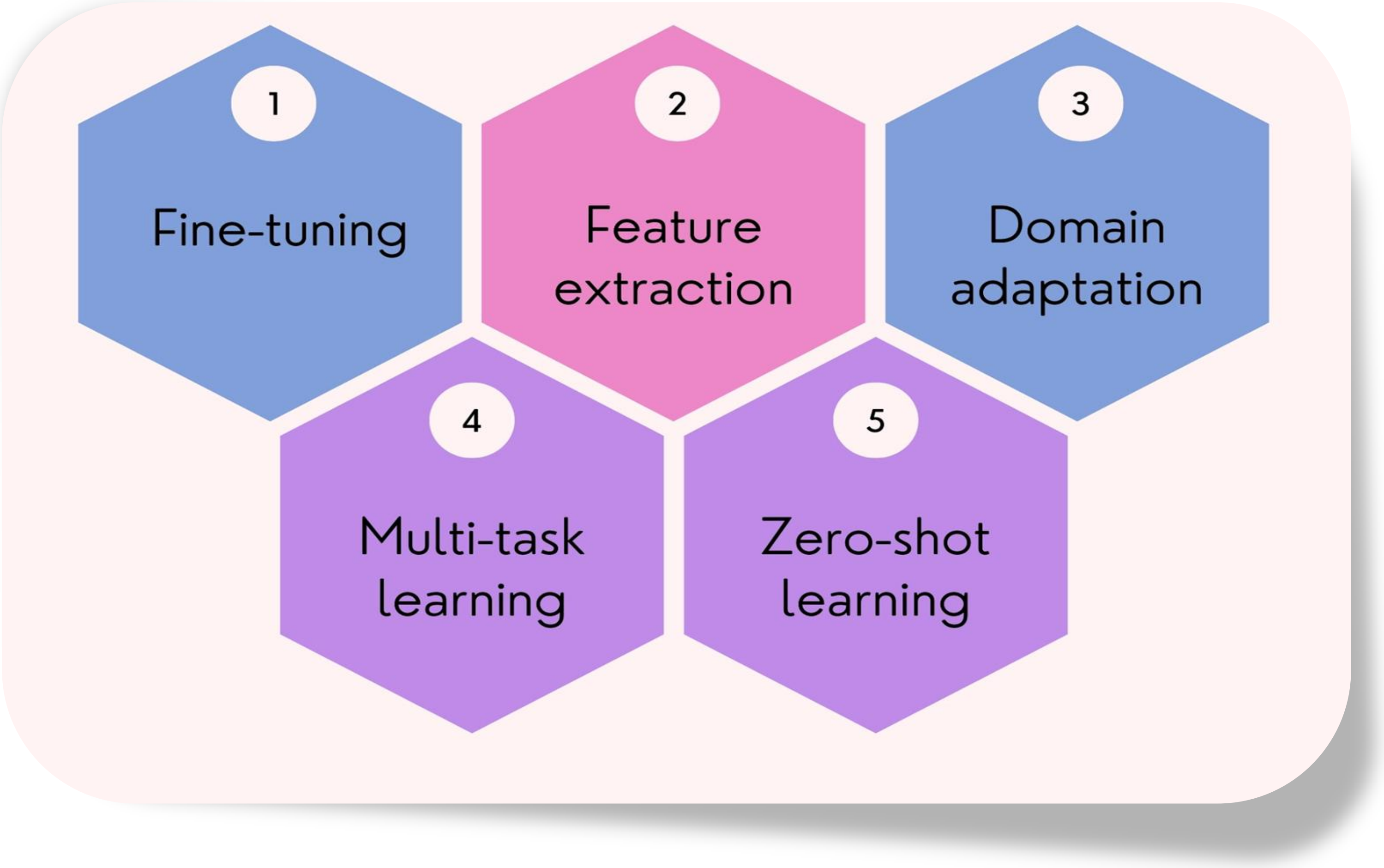


(a) Traditional Machine Learning (b) Transfer Learning

Transductive vs Inductive Transfer Learning

- **Transductive** transfer
  - No labeled target domain data available
  - Focus of most transfer research in NLP
  - E.g. Domain adaptation
- **Inductive** transfer
  - Labeled target domain data available
  - Goal: improve performance on the target task by training on other task(s)
    - Jointly training on >1 task (multi-task learning)
    - Pre-training (e.g. word embeddings)

Types of Transfer Learning



Types of Transfer Learning

Fine-tuning: It involves the use of pre-trained models as the underlying framework and then training on a new task with a lower learning rate.
Feature extraction: Here, the developer uses pre-trained models to extract features from new data. Then they use the best features to train a new classifier.
Domain adaptation: It works by adapting a pre-trained model to a new domain by fine-tuning it on the target domain data.
Multi-task learning: Here, the focus is to train a single model on multiple tasks to improve performance on all tasks.
Zero-shot learning: It involves the use of pre-trained models to make predictions on new classes without any training data for those classes.

Difference between transfer learning and machine learning

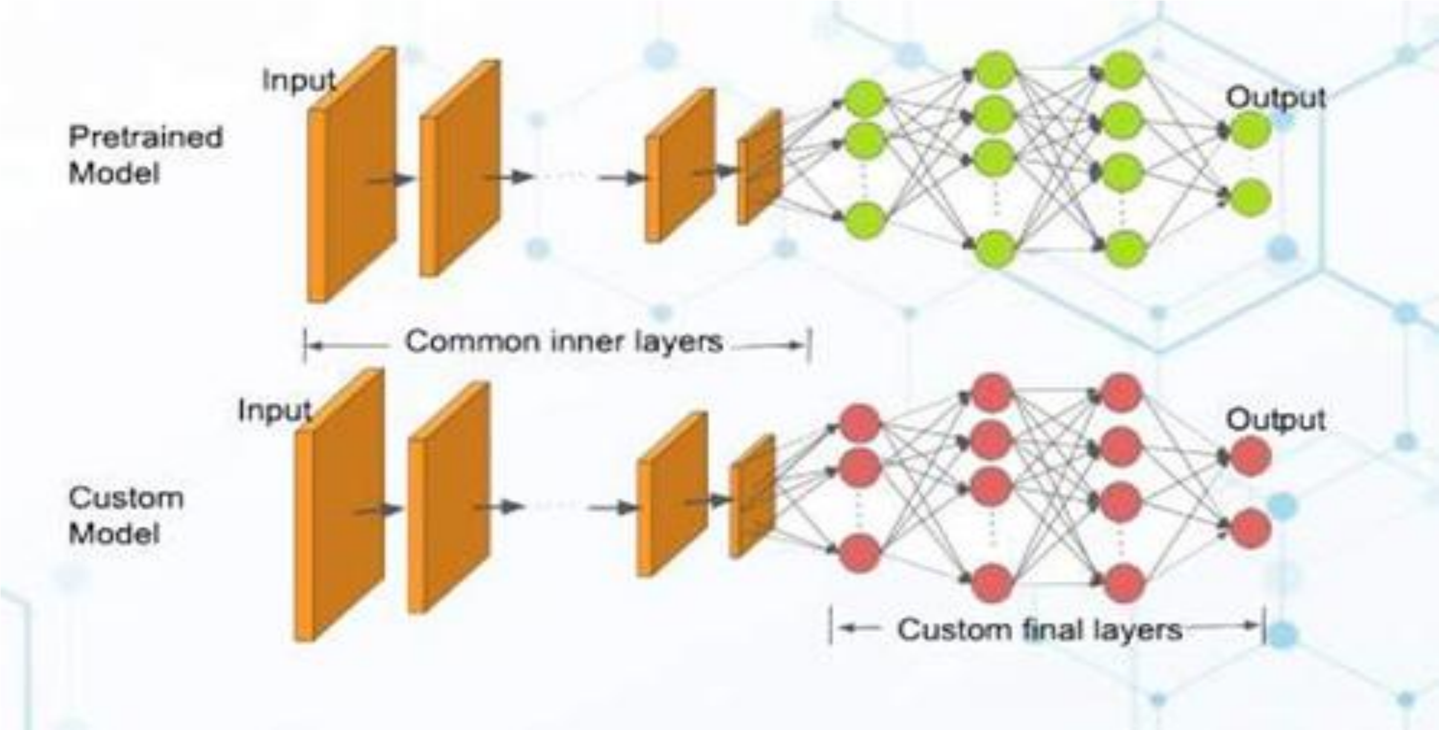
Transfer Learning	**Machine Learning**
This technology focuses on reusing pre-trained models for related tasks	It is a wider domain that encompasses different algorithms and techniques for learning from data
It requires fewer labeled examples for training	Here the developer needs large volumes of labeled data to train models
It ensures faster results and enhanced performance.	The process is a bit slower and hence its implementation needs more time.
Limited to tasks that are related to the source task	Can be applied to a wide range of tasks

Why is Transfer Learning Gaining Popularity?

Data scarcity: Transfer Learning technology doesn’t require reliance on larger data sets. This technology allows models to be fine-tuned using a limited amount of data.
Computational cost: Transfer Learning works on free trade network, thereby reducing the dependency on creating a model from scratch. Thus it is computationally lesser expensive.
Long training time: When the developer starts training a model from scratch, it may take days or weeks. This eventually increases the time. However, in Transfer Learning, the computational time dramatically comes down because it involves the use of pre-trained models. Thus, making it a time-saving process.
Domain adaptation: Transfer Learning enables models to be adapted to new domains by fine-tuning pre-trained models on task-specific data.

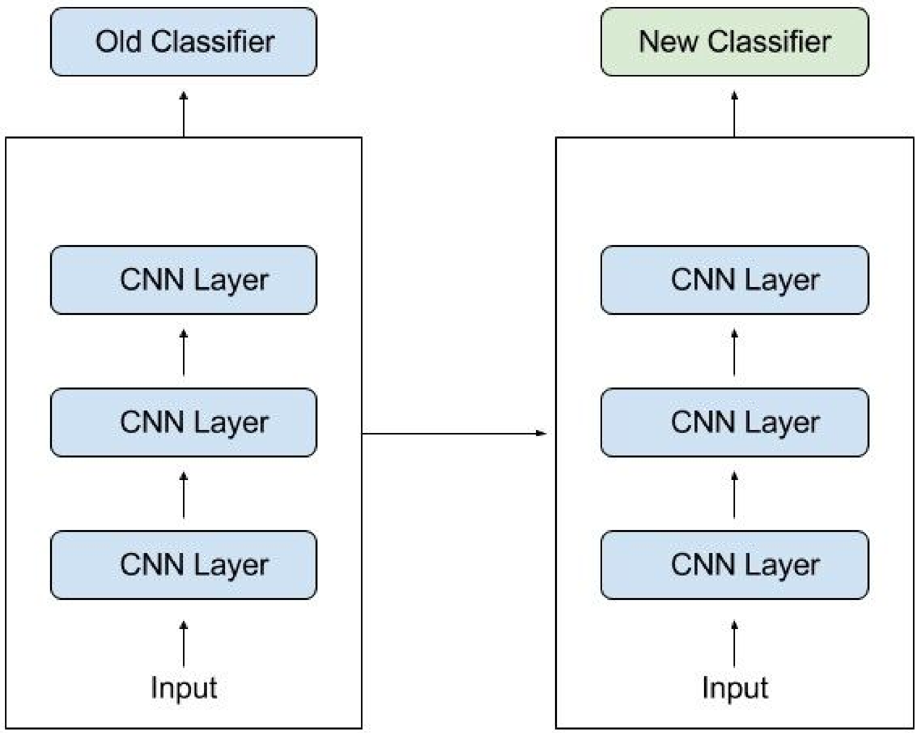
How Transfer Learning Works?

- In computer vision, neural networks typically aim to detect edges in the first layer, forms in the middle layer, and task-specific features in the latter layers.
- The early and central layers are employed in transfer learning, and the latter layers are only retrained. It makes use of the labelled data from the task it was trained on.



Why Should You Use Transfer Learning?

- Transfer learning offers a number of advantages, the most important of which are reduced training time, improved neural network performance (in most circumstances), and the absence of a large amount of data.
- To train a neural model from scratch, a lot of data is typically needed, but access to that data isn’t always possible – this is when transfer learning comes in handy.



Steps in Transfer Learning

1. Start with pre-trained network
2. Partition network into:
    1. Featurizers: identify which layers to keep
    2. Classifiers: identify which layers to replace
3. Re-train classifier layers with new data
4. Unfreeze weights and fine-tune whole network with smaller learning rate

When to finetune your model?

- New dataset is small and similar to original dataset.
  - train a linear classifier on the CNN codes

- New dataset is large and similar to the original dataset
  - fine-tune through the full network

- New dataset is small but very different from the original dataset
  - SVM classifier from activations somewhere earlier in the network

- New dataset is large and very different from the original dataset
  - fine-tune through the entire network

Transfer Learning Services

- Transfer learning is used in many "train your own AI model" services:
- just upload 5-10 images to train a new model! in minutes!

Easily customize your own state-of-the-art computer vision models for your unique use case. Just upload a few labeled images and let Custom Vision Service do the hard work. With just one click, you can export trained models to be run on device or as Docker containers.

Advantages

- Speed up the training process: By using a pre-trained model, the model can learn more quickly and effectively on the second task, as it already has a good understanding of the features and patterns in the data.
- Better performance: Transfer learning can lead to better performance on the second task, as the model can leverage the knowledge it has gained from the first task.
- Handling small datasets: When there is limited data available for the second task, transfer learning can help to prevent overfitting, as the model will have already learned general features that are likely to be useful in the second task.

Disadvantages

- Domain mismatch: The pre-trained model may not be well-suited to the second task if the two tasks are vastly different or the data distribution between the two tasks is very different.
- Overfitting: Transfer learning can lead to overfitting if the model is fine-tuned too much on the second task, as it may learn task-specific features that do not generalize well to new data.
- Complexity: The pre-trained model and the fine-tuning process can be computationally expensive and may require specialized hardware.

Applications

1. Natural language processing (NLP)

Natural language processing refers to a system capable of comprehending and analyzing human language in audio or text files.

2. Computer vision (CV)

Computer vision enables systems to derive meaning from visual data fed through images or videos. ML algorithms train on large datasets (images) and refine themselves to be able to recognize images or classify objects within the images.

3. Neural networks

Neural networks are key to deep learning as they are designed to simulate and replicate human brain functions. Training neural networks require a heavy load of resources due to the complexity of the models they tend to provide.

Applications

3. Healthcare sector

Electromyographic (EMG) signals that evaluate muscle response bear some similarity to electroencephalographic (EEG) brainwaves. As a result, both EMG and EEG signals can use transfer learning to accomplish tasks such as gesture recognition.

4. Emails

Transfer learning can be exploited for an AI model trained to categorize emails to filter out spam (spam filtering).

5. Ecommerce industry

In the ecommerce space, monitoring and tracking customer behavior is crucial for generating more sales. With transfer learning, organizations now have the capability to zero in on subjective customer experiences based on sentiment analysis.

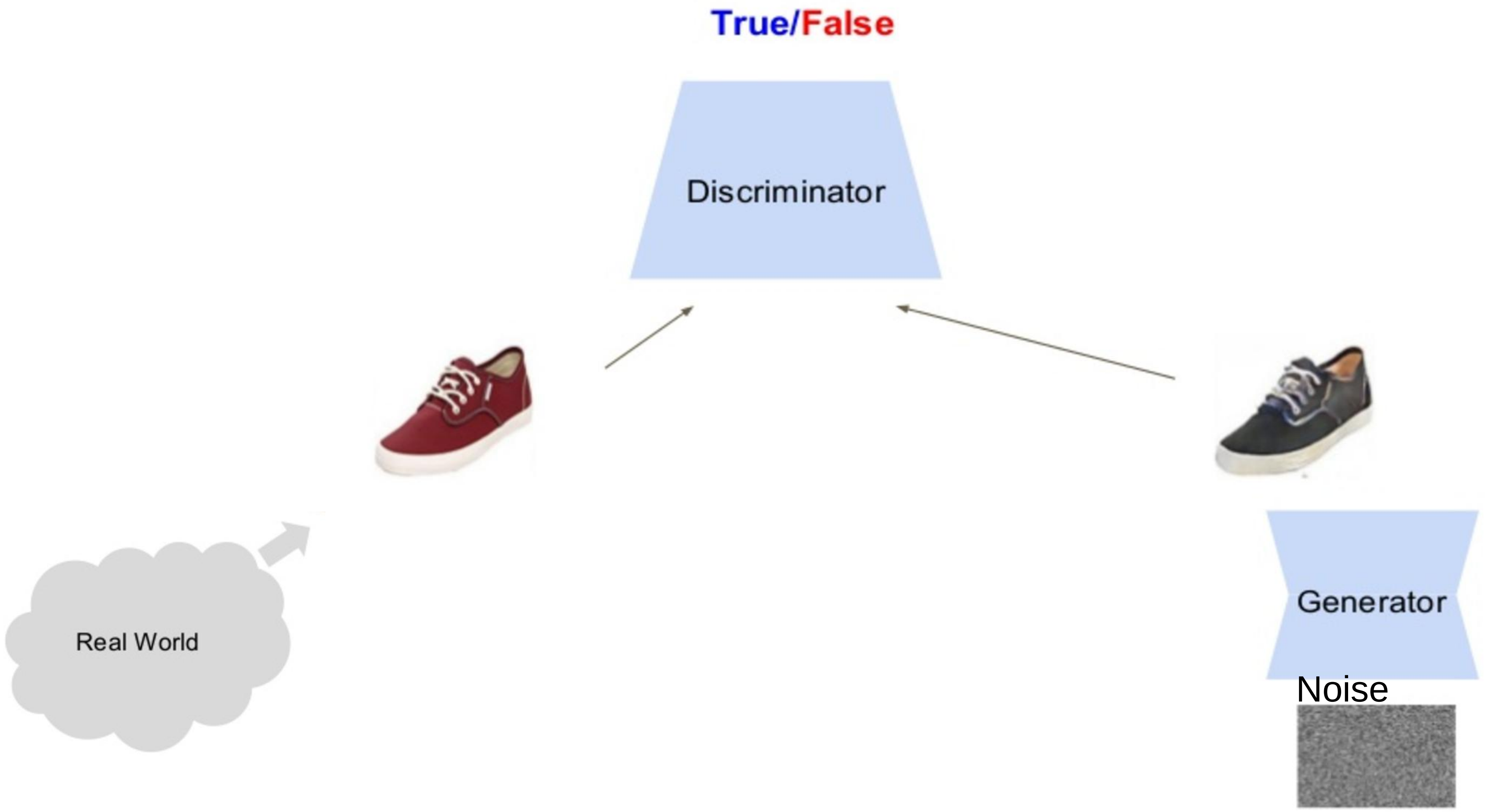
“Thank You”

Generative Adversarial Networks

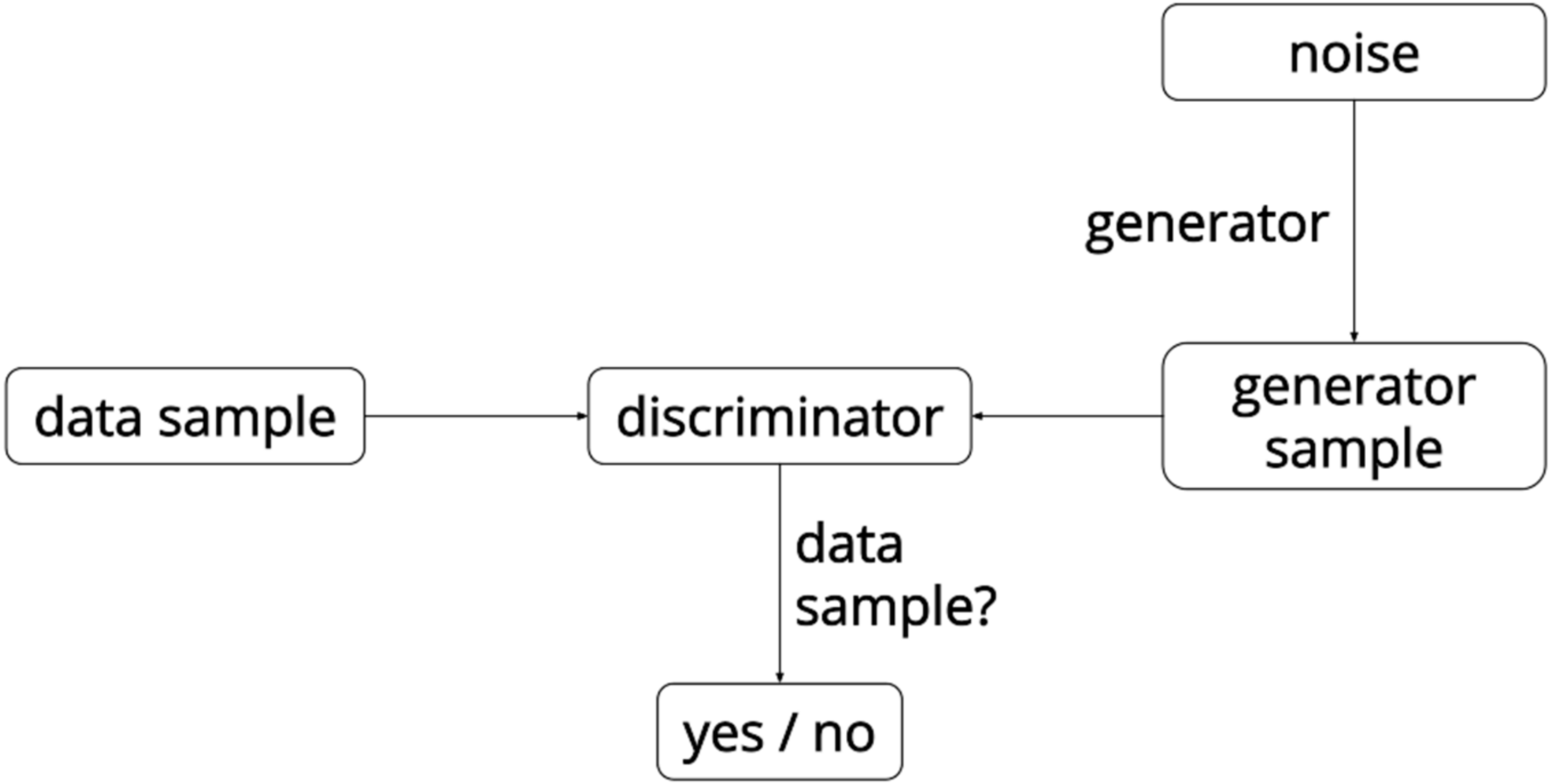
What are GANs?

- System of two neural networks competing against each other in a zero-sum game framework.
- They were first introduced by Ian Goodfellow *et al.* in 2014.
- Can learn to draw samples from a model that is similar to data that we give them.

Components of GANs



Overview of GANs



Discriminative Models

- A **discriminative** model learns a function that maps the input data (x) to some desired output class label (y).

- In probabilistic terms, they directly learn the conditional distribution $P(y|x)$.

Generative Models

- A **generative** model tries to learn the joint probability of the input data and labels simultaneously i.e. *P(x,y)*.

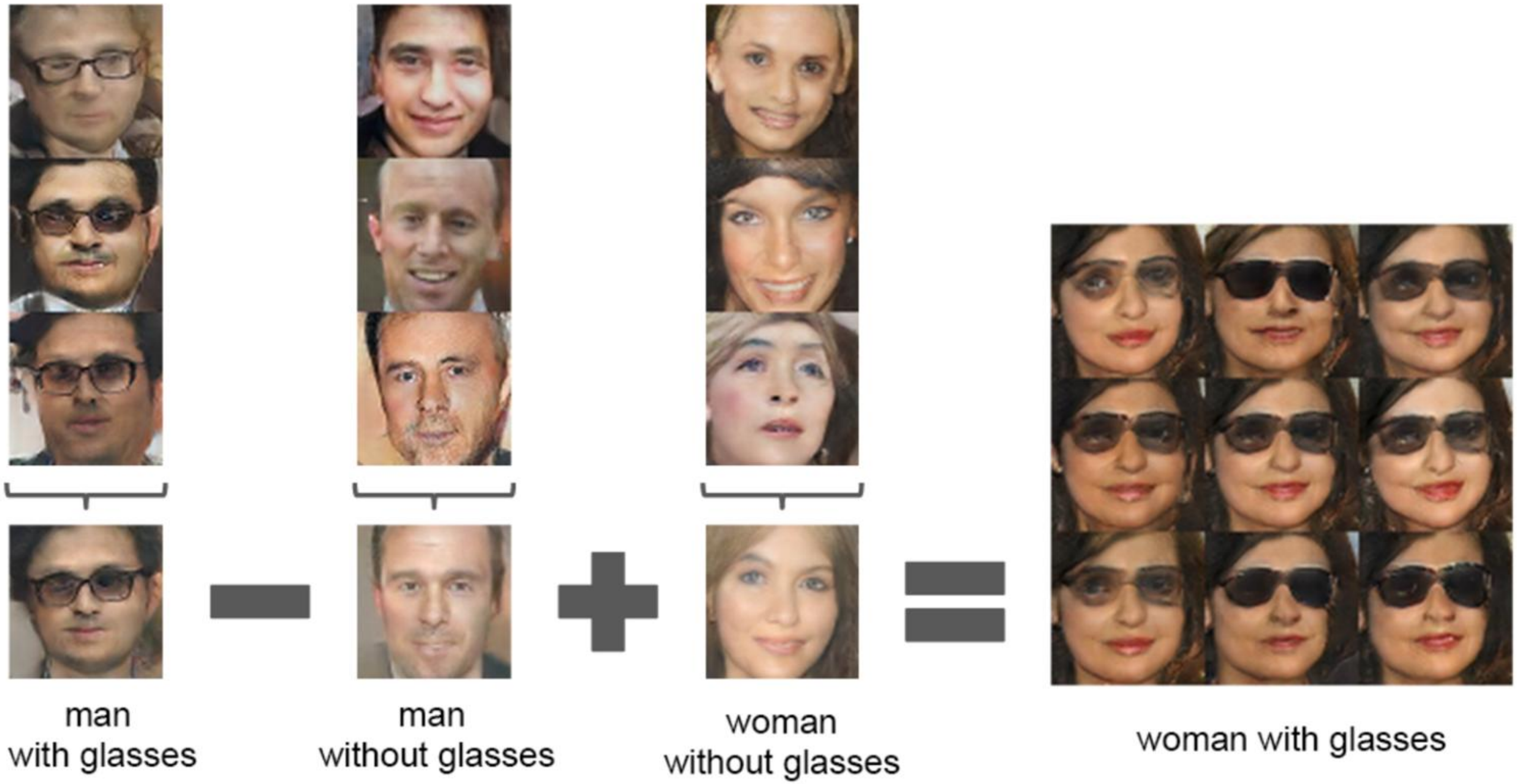
- Potential to understand and explain the underlying structure of the input data even when there are no labels.

How GANs are being used?

- Applied for modelling natural images.
- Performance is fairly good in comparison to other generative models.
- Useful for unsupervised learning tasks.

Why GANs?

- Use a latent code.
- Asymptotically consistent (unlike variational methods) .
- No Markov chains needed.
- Often regarded as producing the best samples.


man
with glasses
man
without glasses
woman
without glasses
woman with glasses

How to train GANs?

- Objective of generative network - increase the error rate of the discriminative network.
- Objective of discriminative network – decrease binary classification loss.
- Discriminator training - backprop from a binary classification loss.
- Generator training - backprop the negation of the binary classification loss of the discriminator.

Loss Functions

$$\mathcal{L}(\hat{x}) = \min_{x \in data} (x - \hat{x})^2$$

Generator

$$D_G^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}$$

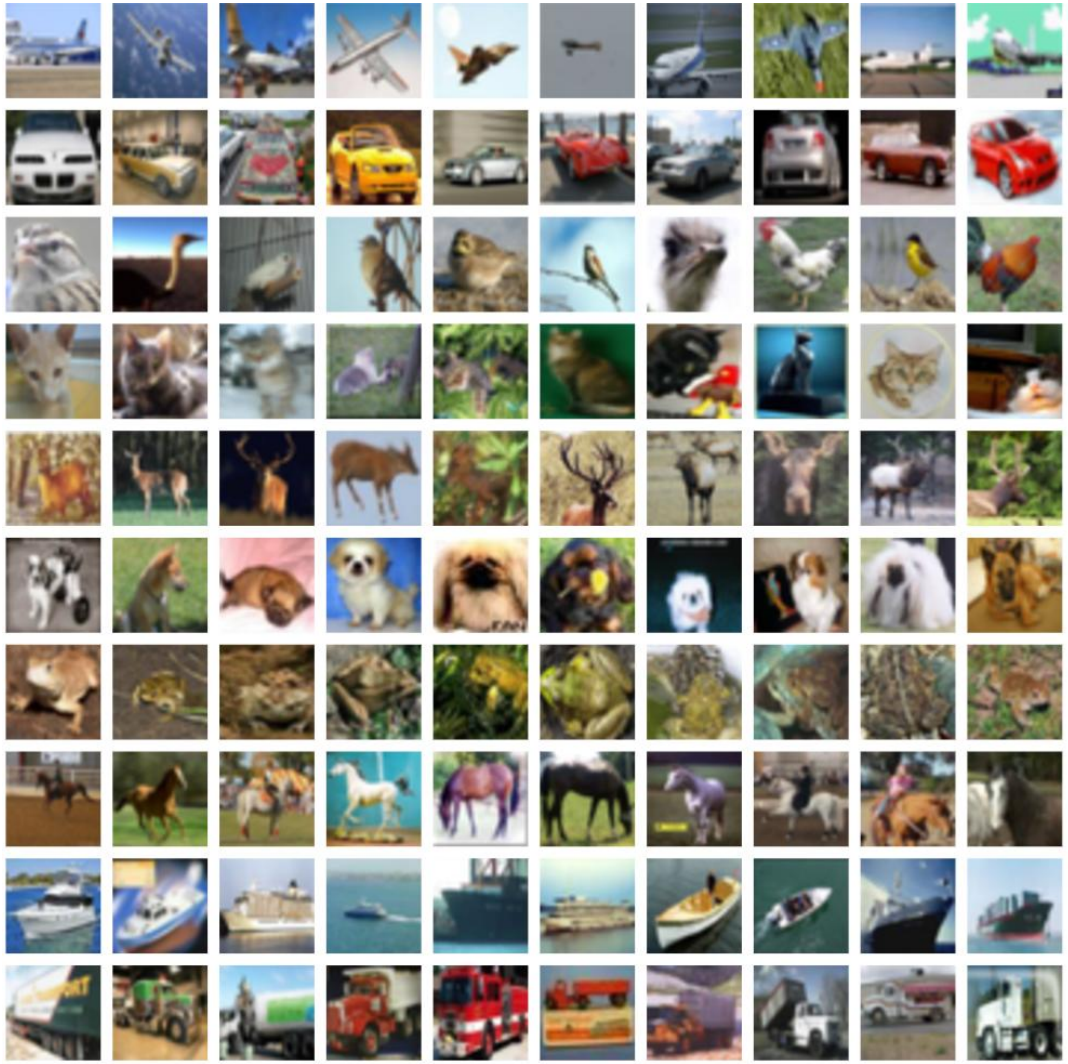
Discriminator

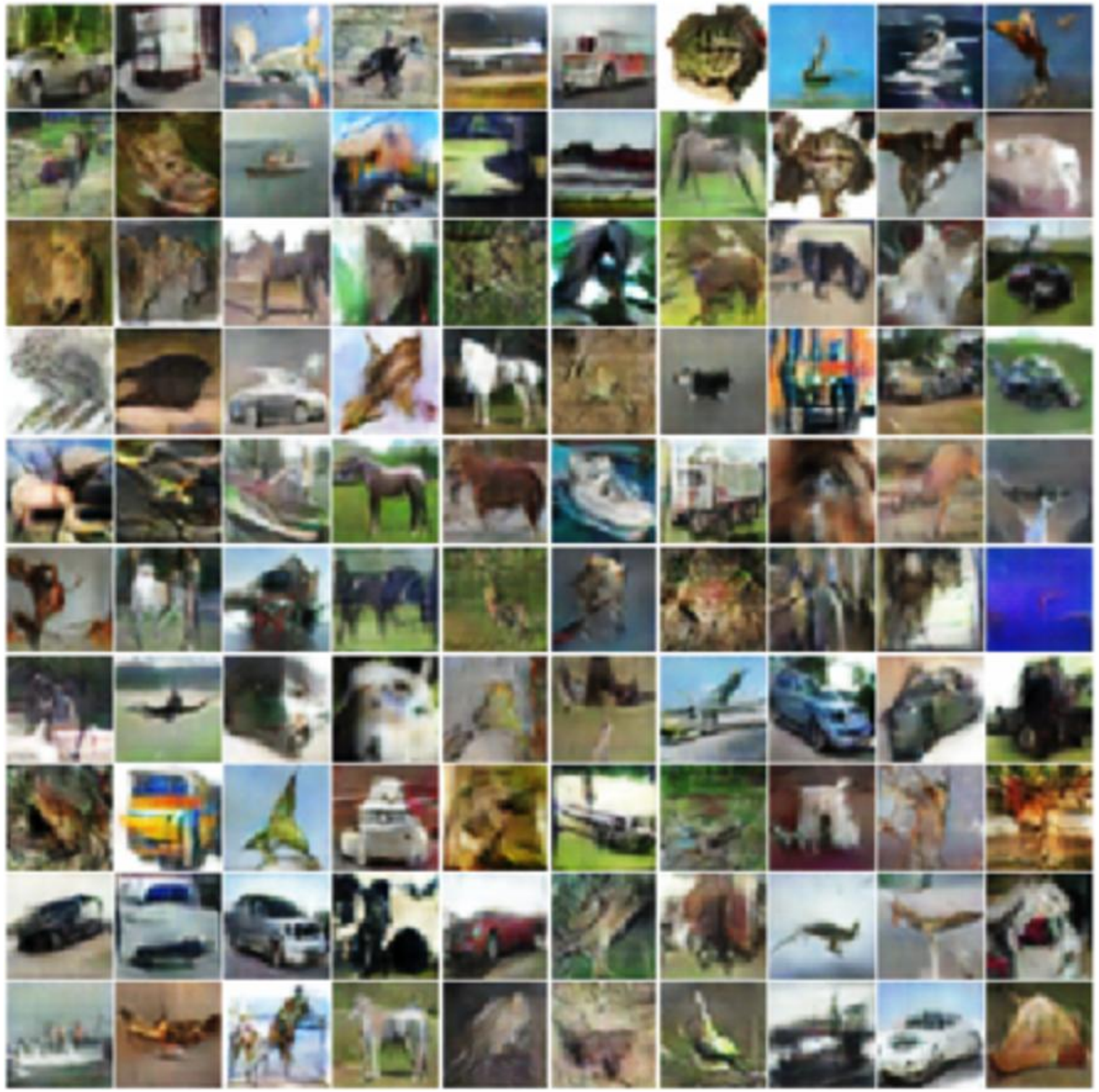
Generated bedrooms. Source: "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks" https://arxiv.org/abs/1511.06434v2

“Improved Techniques for Training GANs” *by* Salimans et. al

- One-sided Label smoothing - replaces the 0 and 1 targets for a classifier with smoothed values, like .9 or .1 to reduce the vulnerability of neural networks to adversarial examples.

- Virtual batch Normalization - each example x is normalized based on the statistics collected on a reference batch of examples that are chosen once and fixed at the start of training, and on x itself.





Original CIFAR-10 vs. Generated CIFAR-10 samples

Source: "Improved Techniques for Training GANs" https://arxiv.org/abs/1606.03498

Variations to GANs

- Several new concepts built on top of GANs have been introduced –
    - InfoGAN – Approximate the data distribution and learn interpretable, useful vector representations of data.
    - Conditional GANs - Able to generate samples taking into account external information (class label, text, another image). Force *G* to generate a particular type of output.

Major Difficulties

- Networks are difficult to converge.
- Ideal goal – Generator and discriminator to reach some desired equilibrium but this is rare.
- GANs are yet to converge on large problems (E.g. Imagenet).

Common Failure Cases

- The discriminator becomes too strong too quickly and the generator ends up not learning anything.
- The generator only learns very specific weaknesses of the discriminator.
- The generator learns only a very small subset of the true data distribution.

So what can we do?

- Normalize the inputs
- A modified loss function
- Use a spherical Z
- BatchNorm
- Avoid Sparse Gradients: ReLU, MaxPool
- Use Soft and Noisy Labels
- DCGAN / Hybrid Models
- Track failures early (D loss goes to 0: failure mode)
- If you have labels, use them
- Add noise to inputs, decay over time

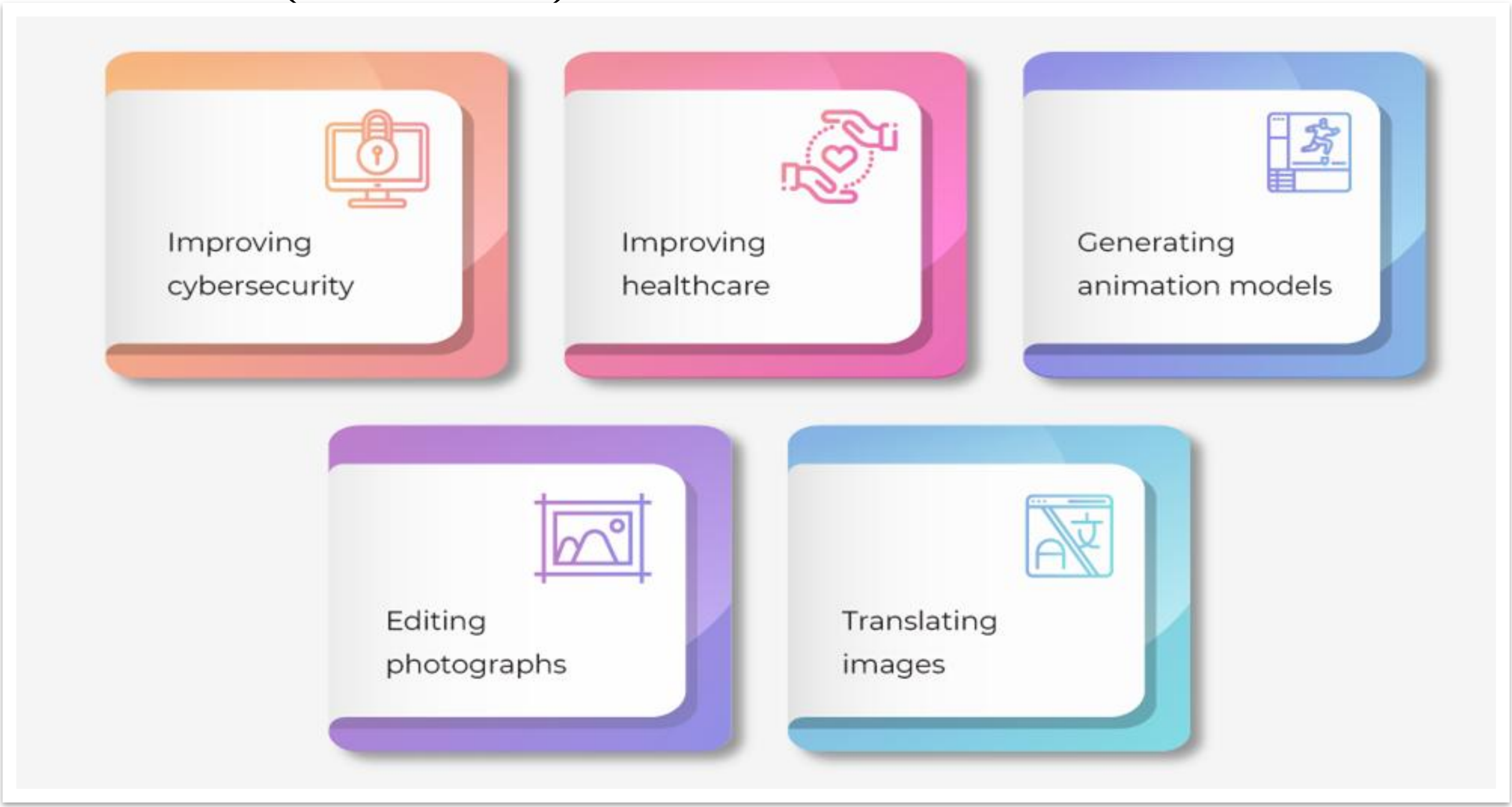
Advantages of Generative Adversarial Networks (GAN’s)

- GANs generate data that looks similar to original data. If you give GAN an image then it will generate a new version of the image which looks similar to the original image. Similarly, it can generate different versions of the text, video, audio.
- GANs go into details of data and can easily interpret into different versions so it is helpful in doing machine learning work.
- By using GANs and machine learning we can easily recognize trees, street, bicyclist, person, and parked cars and also can calculate the distance between different objects.

Disadvantages of Generative Adversarial Networks (GAN’s)

- **Harder to train:** You need to provide different types of data continuously to check if it works accurately or not.
- Generating results from text or speech is very complex

Applications of Generative Adversarial Networks (GAN's)



Conclusions

- Train GAN – Use discriminator as base model for transfer learning and the fine-tuning of a production model.

- A well-trained generator has learned the true data distribution well - Use generator as a source of data that is used to train a production model.

“Thank You”