Deep Learning Algorithms

Deep Learning

- Artificial neural networks are used in deep learning to execute complex computations on enormous volumes of data. It's a sort of machine learning that's based on the human brain's structure and function.
- Industries such as health care, ecommerce, automobiles use the deep learning algorithms to train the machines by learning from examples.

Working of Deep Learning:

- Algorithms leverage unknown elements in the input distribution to extract features, organize objects, and uncover important data patterns throughout the training phase.
- This happens at various levels, employing the algorithms to develop the models, much like training machines for self-learning.

Types of Deep learning algorithms:

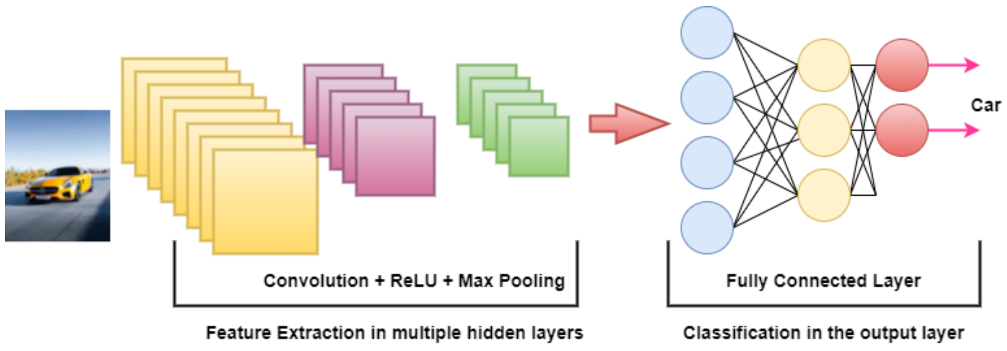
- Convolutional Neural Network
- Long Short-Term Memory Network
- Recurrent Neural Network
- Generative Adversarial Network
- Radial Basis Function Network
- Multilayer Perceptron
- Self-Organizing Maps
- Deep Belief Networks
- Restricted Boltzmann Machines
- Autoencoders

1) Convolutional Neural Networks (CNNs):

- CNNs, also known as ConvNets, are multilayer neural networks that are primarily used for image processing and object detection.
- CNNs are commonly used to detect abnormalities, identify satellite photos, interpret medical imaging, forecast time series, and identify anomalies.

How do CNN work?

- CNN's have multiple layers that process and extract features from data.
    - **Convolution layer** – performs the convolution operation using several filters.
    - **Activation unit** – helps to fire the neurons in the layers.
    - **Pooling layer**- helps to down sample the features.
    - **Flatten layer**- Convert the high dimensional data to single column vector
    - **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.

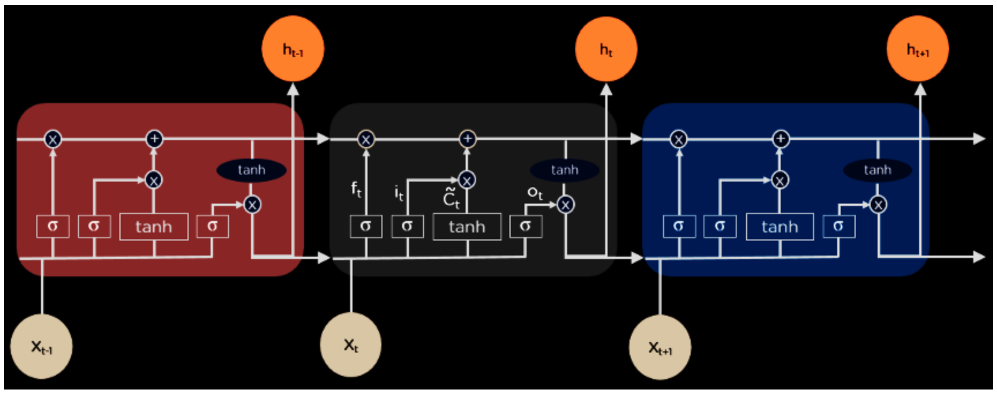

Car
Convolution + ReLU + Max Pooling
Fully Connected Layer
Feature Extraction in multiple hidden layers
Classification in the output layer

2) Long Short-Term Memory (LSTMs):

- Long term dependencies can be learned and remembered using LSTMs, which are a form of Recurrent Neural Network (RNN).
- The default behavior is to recall past information over long periods of time.
- For their nature in remembering previous inputs they are useful in time-series prediction.
- Four interacting layers communicate in a unique way in LSTMs, which have a chain-like structure.
- LSTMs are commonly employed for voice recognition, music creation, and pharmaceutical research, in addition to time-series predictions.

How do LSTM work?

- First, they forget irrelevant parts of the previous state.
- Next, they selectively update the cell-state values.
- Finally, the certain parts of the cell state are outputted.

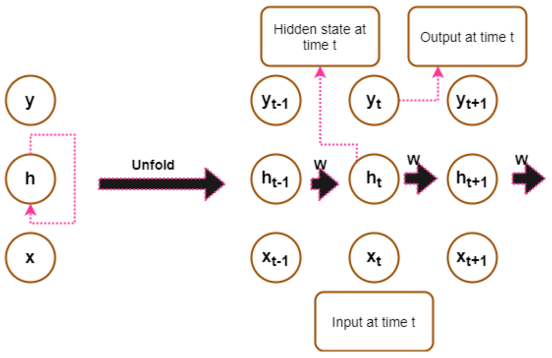


3) Recurrent Neural Networks (RNNs):

- The outputs from the LSTM can be given as inputs to the current phase since RNNs contain connections that create directed cycles.
- The LSTM's output becomes an input to the current phase, and its internal memory allows it to remember prior inputs. Image captioning, time-series analysis, natural-language processing, handwriting identification, and machine translation are all common uses for RNNs

How do RNN work?

- The output at time t-1 is feeds into the input at t.
- In same way the output at time t feeds into the input at t+1
- RNN algorithms handle any length of input data.

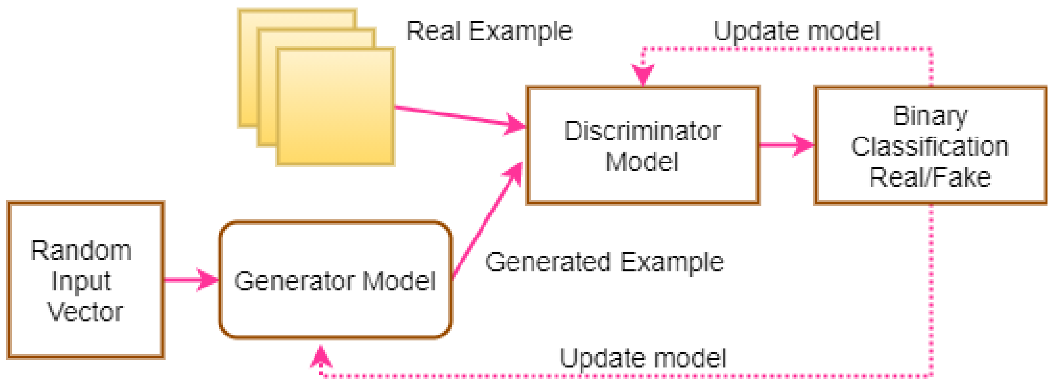


4) Generative Adversarial Networks (GANs):

- GANs are deep learning generative algorithms that generate new data instances that are similar to the training data. GAN is made up of two parts: a generator that learns to generate fake data and a discriminator that learns from that data.
- GANs aid in the creation of realistic images and cartoon characters, as well as the creation of photographs of human faces and the rendering of 3D objects.

How do GAN work?

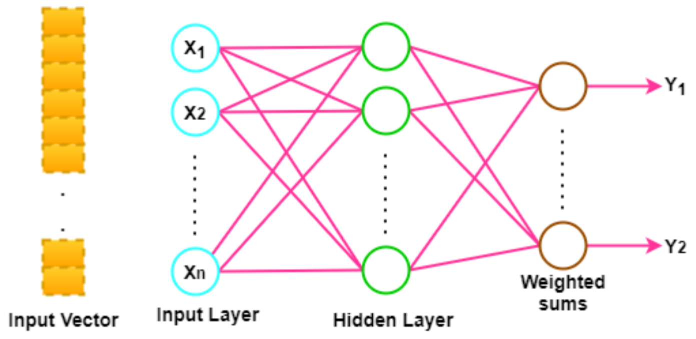
- The discriminator learns to tell the difference between the bogus data generated by the generator and the genuine sample data.
- The generator generates fraudulent data during early training, and the discriminator quickly learns to recognize that it's false.
- To update the model, the GAN provides the results to the generator and discriminator.



5) Radial Basis Function Networks (RBFNs):

- RBFN is used to measure the classification by measuring the similarity between the input and training examples.
- The input vector feeds to the input layer consists of RBF neurons.
- The output layer has one node per category or class of data, and the function finds the weighted total of the inputs.
- The Gaussian transfer functions are found in the neurons in the hidden layer which have outputs that are inversely proportional to the distance from the neuron's center.

- The output of the network is a linear combination of the input's radial-basis functions and the parameters of the neuron.

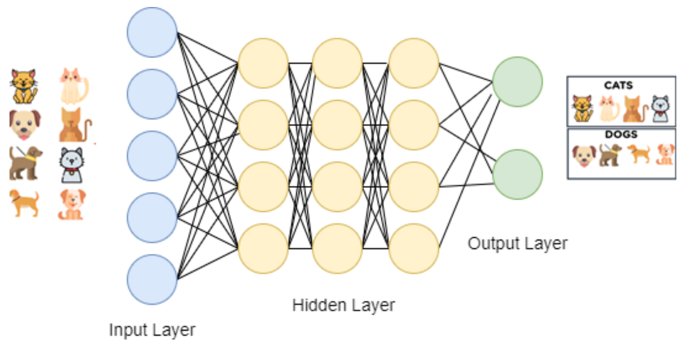


6) Multilayer perceptron (MLPs):

- MLPs are a type of feedforward neural network that consists of multiple layers of perceptron with activation functions.
- MLPs are made up of two fully connected layers: an input layer and an output layer.
- They have the same amount of input and output layers, but they can have several hidden layers, and they can be used to create voice, picture, and machine translation software.

How do MLP work?

- The data feed into the input layer of the network. The signal flows in one way because the layers of neurons are connected in a graph.
- MLP computes the weight between the input and hidden layer.
- Then it uses activation function to determine which nodes to fire. Activation functions includes sigmoid, tanh and ReLU.
- From a training data set, MLPs train the model to grasp the correlation and learn the dependencies between the independent and target variables.

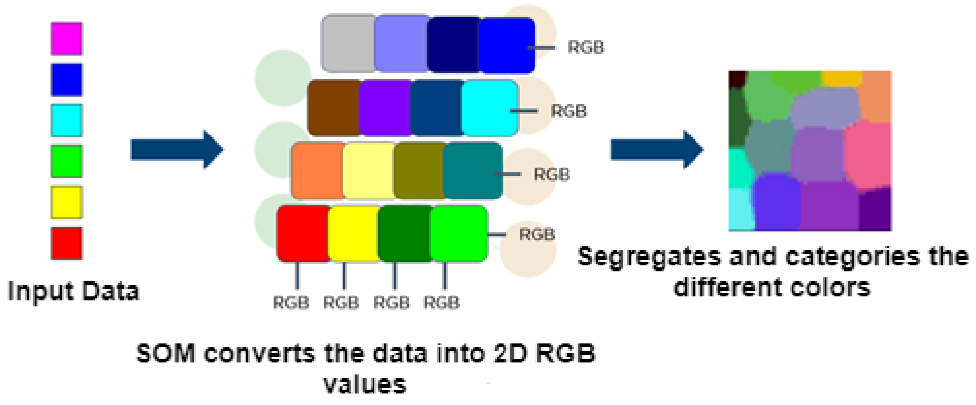


7) Self Organizing Map (SOMs):

- SOM enables the data visualization to reduce the dimensions of data through self-organizing artificial neural networks.
- Data visualization attempts to solve the problem that humans cannot easily visualize in high dimensional data.

How do SOM work?

- SOMs choose a vector at random from the training data to initialize weights for each node.
- SOMs look at each node to see which weights are most likely to be the input vector. The Best Matching Unit (BMU) is the winning node.
- The BMU's neighborhood is discovered through SOMs, and the number of neighbors decreases with time.
- The sample vector is given a winning weight using SOMs. The weight of a node changes as it gets closer to a BMU.
- The further neighbor is from the BMU, the less it learns from it. For N iterations, SOMs repeat step two.


RGB
RGB
RGB
RGB
Input Data
RGB RGB RGB RGB
Segregates and categories the different colors
SOM converts the data into 2D RGB values

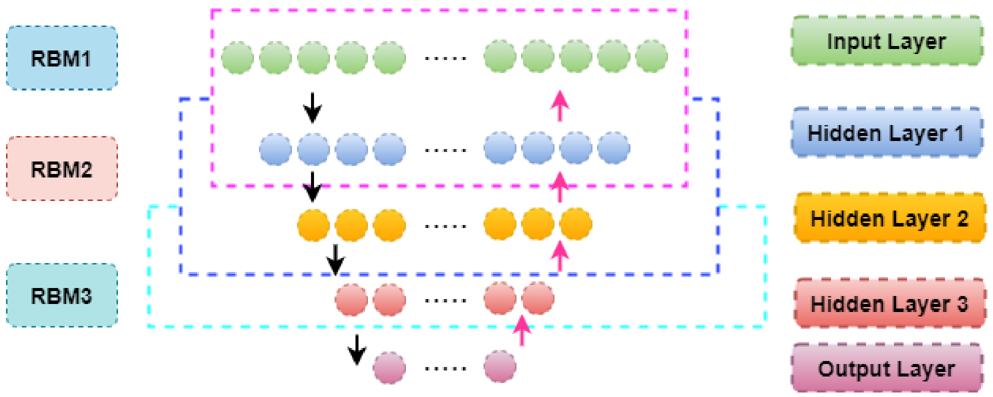
8) Deep Belief Networks (DBNs):

- DBNs are generative models with several layers of stochastic, latent variables. Latent variables, often known as hidden units, have binary values.
- 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 DBN work?

- Greedy learning algorithm used to train the DBN. The algorithm used layer by layer approach for learning top to down generative weights.
- DBN uses the Gibbs sampling on the top two hidden layers and draws the sample from RBM defined by the 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.

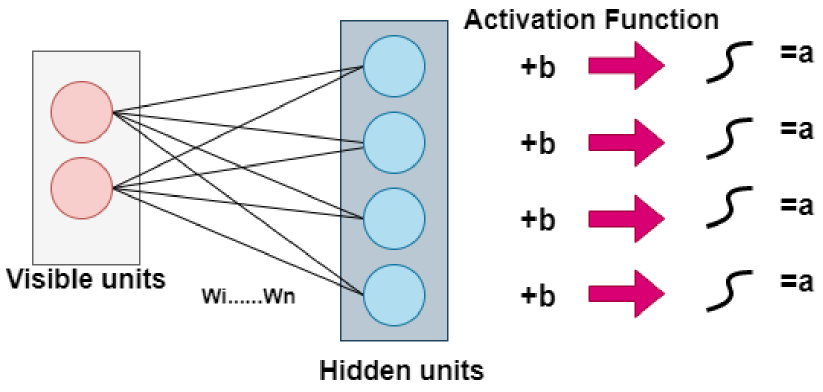


9) Restricted Boltzmann Machines (RBMs):

- BMs are stochastic neural networks that can learn from a probability distribution over a set of inputs.
- RBMs consists of visible units and 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.
- It used in dimensionality reduction, classification, regression, collaborative filtering, feature learning, and topic modeling.

How do RBM work?

- RBMs accept the inputs and translate them into a set of numbers that encodes the inputs in the forward pass.
- Then it combines every input with individual weight and 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.

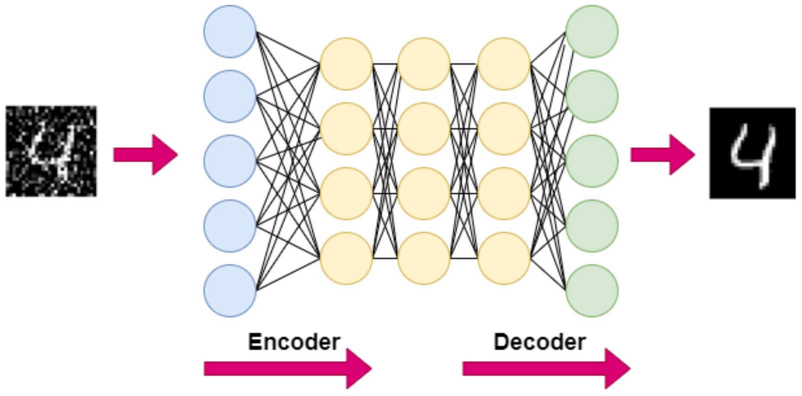


10) Autoencoder:

- Autoencoders are the unique type of feed forward neural network in which input and output are identical.
- The autoencoders are trained to replicate the data from the input layer to the output layer.
- It is widely used in pharma discovery, predicting popularity and imaging application.

How do Autoencoder work?

- Autoencoders are the structure to receive the input and transformed it into the different representation.
- Encoder used to encode the image and then reduce the input size into smaller representation.
- Finally the decoder used to decode the image to generate the reconstructed image.



Important questions:

1. What is pooling on CNN and How does it work?

Pooling is used to reduce the spatial dimensions of a CNN. It performs down-sampling operations to reduce the dimensionality and creates a pooled feature map by sliding a filter matrix over the input matrix.

2. What do you understand by Autoencoder?

Autoencoder is an artificial neural network. It can learn representation for a set of data without any supervision. The network automatically learns by copying its input to the output.

3. What is the biggest problem in GAN?

GANs are difficult to train. The reason they are difficult to train is that both the generator model and the discriminator model are trained simultaneously in a game. This means that improvements to one model come at the expense of the other model.

4. In terms of Dimensionality reduction, how does auto encoder differ from PCA?

- PCA is essentially a linear transformation but Auto-encoders are capable of modelling complex non linear functions.

- PCA features are totally linearly uncorrelated with each other since features are projections onto the orthogonal basis. But autoencoder features might have correlations since they are just trained for accurate reconstruction.
- PCA is faster and computationally cheaper than autoencoders.
- A single layered autoencoder with a linear activation function is very similar to PCA.
- Autoencoder is prone to overfitting due to high number of parameters.

5. Why do we require self-organizing feature map?

The self-organizing map makes topologically ordered mappings between input data and processing elements of the map. Topological ordered implies that if two inputs are of similar characteristics, the most active processing elements answering to inputs that are located closed to each other on the map.

6. What is the difference between Autoencoders and RBMs?

RBMs are generative. That is, unlike autoencoders that only discriminate some data vectors in favour of others, RBMs can also generate new data with given joined distribution. They are also considered more feature-rich and flexible.

7. What are some of the Deep Learning frameworks or tools available in market?

Some of the top Deep Learning frameworks out there today are:

- TensorFlow
- Keras
- PyTorch
- Caffe2
- CNTK
- MXNet
- Theano

8. What is the meaning of valid padding and same padding in CNN?

- Valid padding: It is used when there is no requirement for padding. The output matrix will have the dimensions (n – f + 1) X (n – f + 1) after convolution.
- Same padding: Here, padding elements are added all around the output matrix. It will have the same dimensions as the input matrix.

9. Why is zero initialization not a good weight initialization process?

- If the set of weights in the network is put to a zero, then all the neurons at each layer will start producing the same output and the same gradients during backpropagation.
- As a result, the network cannot learn at all because there is no source of asymmetry between neurons. That is the reason why we need to add randomness to the weight initialization process.

10. What is matrix element-wise multiplication? Explain with an example.

Element-wise matrix multiplication is used to take two matrices of the same dimensions. It further produces another combined matrix with the elements that are a product of corresponding elements of matrix a and b.

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \circ \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix} =$$

$$\begin{pmatrix} a_{11}b_{11} & a_{12}b_{12} & a_{13}b_{13} \\ a_{21}b_{21} & a_{22}b_{22} & a_{23}b_{23} \\ a_{31}b_{31} & a_{32}b_{32} & a_{33}b_{33} \end{pmatrix}$$

11. What are the issues faced while training in Recurrent Networks?

Recurrent Neural Network uses backpropagation algorithm for training, but it is applied on every timestamp. It is usually known **as Back-propagation Through Time** (BTT).

There are two significant issues with Back-propagation, such as:

1) Vanishing Gradient
2) Exploding Gradient

12. What is Data Normalization, and why do we need it?

Data normalization is used during backpropagation. The main motive behind data normalization is to reduce or eliminate data redundancy. Here we rescale values to fit into a specific range to achieve better convergence.

13. What are Type 1 and Type 2 errors?

Type 1 error is when your algorithm makes a positive prediction, but in fact, it's negative. For example, your algorithm predicted a patient has diabetes, but in fact, he doesn't.
Type 2 error is when your algorithm makes a negative prediction, but in fact, it's positive. For example, your algorithm predicted a patient doesn't have diabetes, but in fact, they do.

14. What is a model learning rate? Is a high learning rate always good?

The learning rate is a tuning parameter that determines the step size of each iteration (epoch) during model training.

- Your model will converge quickly if the learning rate is high, and the model weights are updated often, but it may overshoot the true error minima. This results in a model that is faster yet incorrect.
- Your model will take a long time to converge but will not overshoot the genuine error minima if the learning rate is low and the model weights are updated slowly. This results in a model that is slower but more accurate.