DEEP LEARNING

B.TECH-IT VIII SEM

QUESTION BANK

Question - What are the applications of Machine Learning .When it is used.

Answer - Artificial Intelligence (AI) is everywhere. One of the popular applications of AI is Machine Learning (ML), in which computers, software, and devices perform via cognition (very similar to human brain). we share few examples of machine learning that we use everyday and perhaps have no idea that they are driven by ML.

1. Virtual Personal Assistants

Siri, Alexa, Google Now are some of the popular examples of virtual personal assistants. As the name suggests, they assist in finding information, when asked over voice. All you need to do is activate them and ask “What is my schedule for today?”, “What are the flights from Germany to London”, or similar questions. For answering, your personal assistant looks out for the information, recalls your related queries, or send a command to other resources (like phone apps) to collect info. You can even instruct assistants for certain tasks like “Set an alarm for 6 AM next morning”, “Remind me to visit Visa Office day after tomorrow”.

Machine learning is an important part of these personal assistants as they collect and refine the information on the basis of your previous involvement with them. Later, this set of data is utilized to render results that are tailored to your preferences.

Virtual Assistants are integrated to a variety of platforms. For example:

- Smart Speakers: Amazon Echo and Google Home
- Smartphones: Samsung Bixby on Samsung S8

- Mobile Apps: Google Allo

2. Predictions while Commuting

Traffic Predictions: We all have been using GPS navigation services. While we do that, our current locations and velocities are being saved at a central server for managing traffic. This data is then used to build a map of current traffic. While this helps in preventing the traffic and does congestion analysis, the underlying problem is that there are less number of cars that are equipped with GPS. Machine learning in such scenarios helps to estimate the regions where congestion can be found on the basis of daily experiences.

Online Transportation Networks: When booking a cab, the app estimates the price of the ride. When sharing these services, how do they minimize the detours? The answer is machine learning. Jeff Schneider, the engineering lead at Uber ATC reveals in a an interview that they use ML to define price surge hours by predicting the rider demand. In the entire cycle of the services, ML is playing a major role.

3. Videos Surveillance

Imagine a single person monitoring multiple video cameras! Certainly, a difficult job to do and boring as well. This is why the idea of training computers to do this job makes sense.

The video surveillance system nowadays are powered by AI that makes it possible to detect crime before they happen. They track unusual behaviour of people like standing motionless for a long time, stumbling, or napping on benches etc. The system can thus give an alert to human attendants, which can ultimately help to avoid mishaps. And when such activities are reported and counted to be true, they help to improve the surveillance services. This happens with machine learning doing its job at the backend.

4. Social Media Services

From personalizing your news feed to better ads targeting, social media platforms are utilizing machine learning for their own and user benefits. Here are a few examples that you must be noticing, using, and loving in your social media accounts, without realizing that these wonderful features are nothing but the applications of ML.

- *People You May Know*: Machine learning works on a simple concept: understanding with experiences. Facebook continuously notices the friends that you connect with, the profiles that you visit very often, your interests, workplace, or a group that you share with someone etc. On the basis of continuous learning, a list of Facebook users are suggested that you can become friends with.
- *Face Recognition*: You upload a picture of you with a friend and Facebook instantly recognizes that friend. Facebook checks the poses and projections in the picture, notice the unique features, and then match them with the people in your friend list. The entire process at the backend is complicated and takes care of the precision factor but seems to be a simple application of ML at the front end.
- *Similar Pins*: Machine learning is the core element of Computer Vision, which is a technique to extract useful information from images and videos. Pinterest uses computer vision to identify the objects (or pins) in the images and recommend similar pins accordingly.

5. Email Spam and Malware Filtering

- There are a number of spam filtering approaches that email clients use. To ascertain that these spam filters are continuously updated, they are powered by machine learning. When rule-based spam filtering is done, it fails to track the latest tricks adopted by spammers. Multi Layer Perceptron, C 4.5 Decision Tree Induction are some of the spam filtering techniques that are powered by ML.
- Over 325, 000 malwares are detected everyday and each piece of code is 90–98% similar to its previous versions. The system security programs that are powered by machine learning understand the coding pattern. Therefore, they detects new malware with 2–10% variation easily and offer protection against them.

6. Online Customer Support

A number of websites nowadays offer the option to chat with customer support representative while they are navigating within the site. However, not every website has a live executive to answer your queries. In most of the cases, you talk to a chatbot. These bots tend to extract information from the website and present it to the customers. Meanwhile, the chatbots advances with time. They tend to understand the user queries better and serve them with better answers, which is possible due to its machine learning algorithms.

7. Search Engine Result Refining

Google and other search engines use machine learning to improve the search results for you. Every time you execute a search, the algorithms at the backend keep a watch at how you respond to the results. If you open the top results and stay on the web page for long, the search engine assumes that the the results it displayed were in accordance to the query. Similarly, if you reach the second or third page of the search results but do not open any of the results, the search engine estimates that the results served did not match requirement. This way, the algorithms working at the backend improve the search results.

8. Product Recommendations

You shopped for a product online few days back and then you keep receiving emails for shopping suggestions. If not this, then you might have noticed that the shopping website or the app recommends you some items that somehow matches with your taste. Certainly, this refines the shopping experience but did you know that it's machine learning doing the magic for you? On the basis of your behaviour with the website/app, past purchases, items liked or added to cart, brand preferences etc., the product recommendations are made.

9. Online Fraud Detection

Machine learning is proving its potential to make cyberspace a secure place and tracking monetary frauds online is one of its examples. For example: Paypal is using ML for protection against money laundering. The company uses a set of tools that helps them to compare millions of transactions taking place and distinguish between legitimate or illegitimate transactions taking place between the buyers and sellers.

Question – Draw and explain the architecture of convolutional network .

Answer -

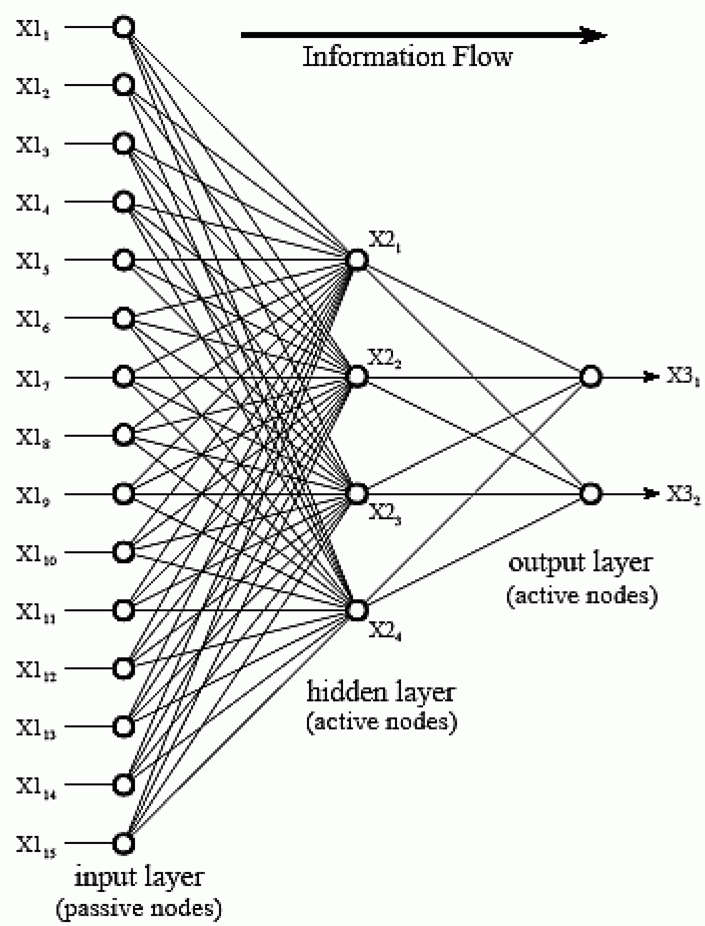


FIGURE 26-5
Neural network architecture. This is the most common structure for neural networks: three layers with full interconnection. The input layer nodes are passive, doing nothing but relaying the values from their single input to their multiple outputs. In comparison, the nodes of the hidden and output layers are active, modifying the signals in accordance with Fig. 26-6. The action of this neural network is determined by the weights applied in the hidden and output nodes.

A Convolutional Neural Network (CNN) is a deep learning algorithm that can recognize and classify features in images for computer vision. It is a multi-layer neural network designed to analyze visual inputs and perform tasks such as image classification, segmentation and object detection, which can be useful for autonomous vehicles. CNNs can also be used for deep learning applications in healthcare, such as medical imaging.

There are two main parts to a CNN:

- A convolution tool that splits the various features of the image for analysis
- A fully connected layer that uses the output of the convolution layer to predict the best description for the image.

link to deep learning in healthcare article

Basic Convolutional Neural Network Architecture

CNN architecture is inspired by the organization and functionality of the visual cortex and designed to mimic the connectivity pattern of neurons within the human brain.

The neurons within a CNN are split into a three-dimensional structure, with each set of neurons analyzing a small region or feature of the image. In other words, each group of neurons specializes in identifying one part of the image. CNNs use the predictions from the layers to produce a final output that presents a vector of probability scores to represent the likelihood that a specific feature belongs to a certain class.

How a Convolutional Neural Network Works—The CNN layers

A CNN is composed of several kinds of layers:

- **Convolutional layer**—creates a feature map to predict the class probabilities for each feature by applying a filter that scans the whole image, few pixels at a time.
- **Pooling layer (downsampling)**—scales down the amount of information the convolutional layer generated for each feature and maintains the most essential information (the process of the convolutional and pooling layers usually repeats several times).

- **Fully connected input layer**—“flattens” the outputs generated by previous layers to turn them into a single vector that can be used as an input for the next layer.
- **Fully connected layer**—applies weights over the input generated by the feature analysis to predict an accurate label.
- **Fully connected output layer**—generates the final probabilities to determine a class for the image.

Popular Convolutional Neural Network Architectures

The architecture of a CNN is a key factor in determining its performance and efficiency. The way in which the layers are structured, which elements are used in each layer and how they are designed will often affect the speed and accuracy with which it can perform various tasks.

The ImageNet Challenge

The ImageNet project is a visual database designed for use in the research of visual object recognition software. The ImageNet project has more than 14 million images specifically designed for training CNN in object detection, one million of which also provide bounding boxes for the use of networks such as YOLO.

Since 2010, the project hosts an annual contest called the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). The contenders of the contest build software programs that attempt to correctly detect and classify objects and scenes within the given images. Currently, the challenge uses a cut down list of a thousand separate classes.

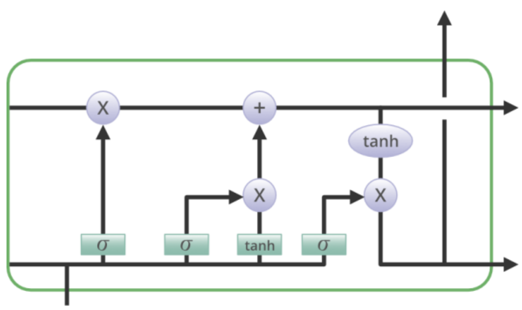
When the annual ILSVRC competition began, a good classification rate was 25%, the first major leap in performance was achieved by a network called AlexNet in 2012, which dropped the classification rate by 10%. Over the next years, the error rates dropped to lower percentages and finally exceeded human capabilitie

Question – Explain LSTM (Long Short Term Memory)

Answer - Long Short Term Memory is a kind of recurrent neural network. In RNN output from the last step is fed as input in the current step. LSTM was desgined by Hochreiter&Schmidhuber. It tackled the problem of long-term dependencies of RNN in which the RNN cannot predict the word stored in the long term memory but can give more accurate predictions from the recent information. As the gap length increases RNN does not give efficent performance. LSTM can by default retain the information for long period of time. It is used for processing, predicting and classifying on the basis of time series data.

Structure Of LSTM:

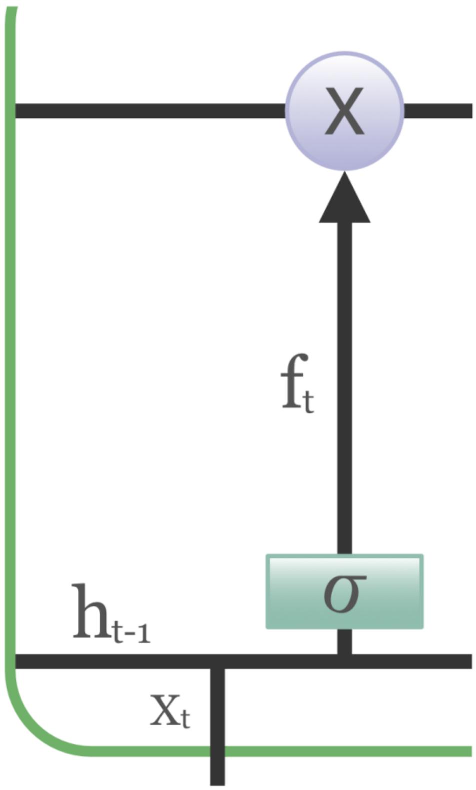
LSTM has a chain structure that contains four neural networks and different memory blocks called **cells**.



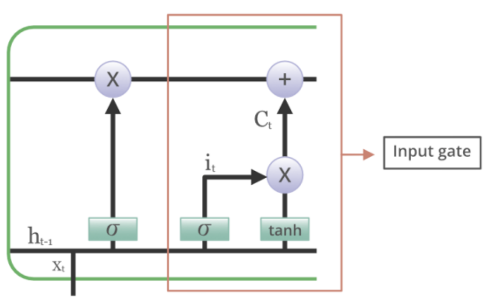
Information is retained by the cells and the memory manipulations are done by the **gates.** There are three gates –

1 - **Forget Gate:** The information that no longer useful in the cell state is removed with the forget gate. Two inputs *x_t* (input at the particular time) and *h_t-1* (previous cell output) are fed to the gate and multiplied with weight matrices followed by the addition of bias. The resultant is passed through an activation function which gives a binary output. If for a particular cell state the output is 0, the piece of information is forgotten and for the output 1, the information is retained for the future use.

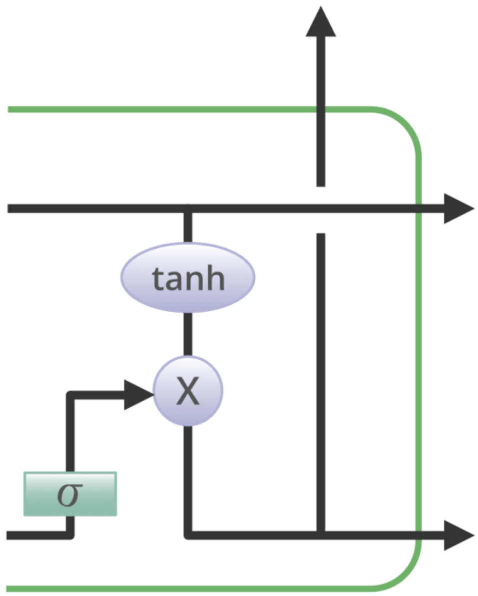
2 –



2 - **Input gate:** Addition of useful information to the cell state is done by input gate. First, the information is regulated using the sigmoid function and filter the values to be remembered similar to the forget gate using inputs *h_t-1* and *x_t*. Then, a vector is created using *tanh* function that gives output from -1 to +1, which contains all the possible values from h_t-1 and *x_t*. Atlast, the values of the vector and the regulated values are multiplied to obtain the useful information



3 - Output gate: The task of extracting useful information from the current cell state to be presented as an output is done by output gate. First, a vector is generated by applying tanh function on the cell. Then, the information is regulated using the sigmoid function and filter the values to be remembered using inputs *h_t-1* and *x_t*. Atlast, the values of the vector and the regulated values are multiplied to be sent as an output and input to the next cell.



Some of the famous applications of LSTM includes:

1. Language Modelling
2. Machine Translation
3. Image Captioning
4. Handwriting generation
5. Question Answering Chatbots

Question – Difference between Deep and Shallow Network.

Answer - Besides an input layer and an output layer, a neural network has intermediate layers, which might also be called hidden layers. They might also be called encoders.

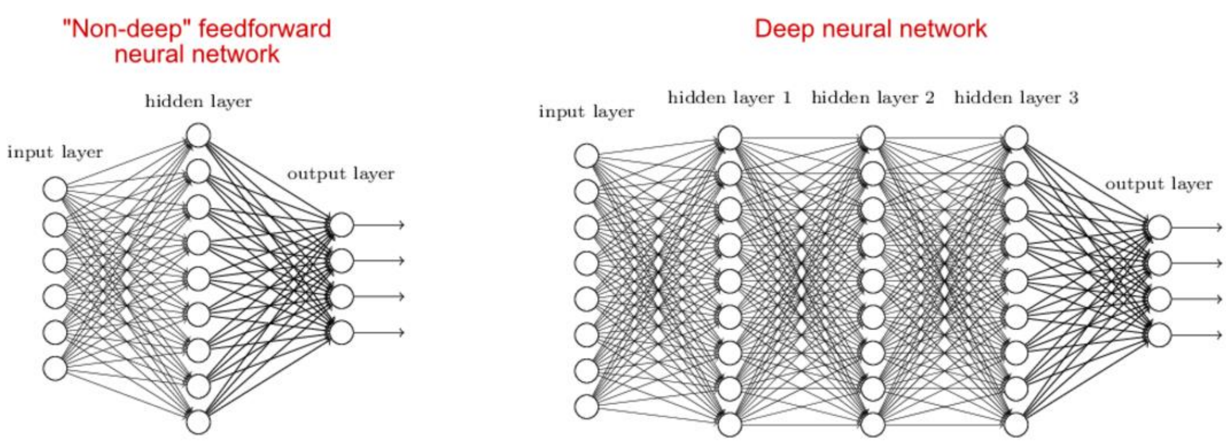
A shallow network has less number of hidden layers. While there are studies that a shallow network can fit any function, it will need to be really fat. That causes the number of parameters to increase a lot.

There are quite conclusive results that a deep network can fit functions better with less parameters than a shallow network.

In short, "shallow" neural networks is a term used to describe NN that usually have only one hidden layer as opposed to deep NN which have several hidden layers, often of various types.

There are papers that highlight that deep NN with the right architectures achieve better results than shallow ones that have the same computational power (e.g. number of neurons or connections).

The main explanation is that the deep models are able to extract/build better features than shallow models and to achieve this they are using the intermediate hidden layers.



Question – What is deep learning , Explain its uses and application and history.

Answer - **Deep learning** is a subset of **machine learning** where artificial **neural networks**, algorithms inspired by the human brain, **learn** from large amounts of data. ... **Deep learning** allows machines to solve complex problems even when using a data set that is very diverse, unstructured and inter-connected.

Uses and Application of Deep Learning

1. Automatic Colorization of Black and White Images

Image colorization is the problem of adding color to black and white photographs.

Traditionally this was done by hand with human effort because it is such a difficult task.

Deep learning can be used to use the objects and their context within the photograph to color the image, much like a human operator might approach the problem.

A visual and highly impressive feat.

This capability leverages of the high quality and very large convolutional neural networks trained for ImageNet and co-opted for the problem of image colorization.

Generally the approach involves the use of very large convolutional neural networks and supervised layers that recreate the image with the addition of color.

2. Automatically Adding Sounds To Silent Movies

In this task the system must synthesize sounds to match a silent video.

The system is trained using 1000 examples of video with sound of a drum stick striking different surfaces and creating different sounds. A deep learning model associates the video frames with a database of pre-rerecorded sounds in order to select a sound to play that best matches what is happening in the scene.

The system was then evaluated using a turing-test like setup where humans had to determine which video had the real or the fake (synthesized) sounds.

A very cool application of both convolutional neural networks and LSTM recurrent neural networks.

3. Automatic Machine Translation

This is a task where given words, phrase or sentence in one language, automatically translate it into another language.

Automatic machine translation has been around for a long time, but deep learning is achieving top results in two specific areas:

- Automatic Translation of Text.
- Automatic Translation of Images.

Text translation can be performed without any preprocessing of the sequence, allowing the algorithm to learn the dependencies between words and their mapping to a new language. Stacked networks of large LSTM recurrent neural networks are used to perform this translation.

As you would expect, convolutional neural networks are used to identify images that have letters and where the letters are in the scene. Once identified, they can be turned into text, translated and the image recreated with the translated text. This is often called instant visual translation.

4. Object Classification and Detection in Photographs

This task requires the classification of objects within a photograph as one of a set of previously known objects.

State-of-the-art results have been achieved on benchmark examples of this problem using very large convolutional neural networks. A breakthrough in this problem by Alex Krizhevsky et al. results on the ImageNet classification problem called AlexNet.

5. Automatic Handwriting Generation

This is a task where given a corpus of handwriting examples, generate new handwriting for a given word or phrase.

The handwriting is provided as a sequence of coordinates used by a pen when the handwriting samples were created. From this corpus the relationship between the pen movement and the letters is learned and new examples can be generated ad hoc.

What is fascinating is that different styles can be learned and then mimicked. I would love to see this work combined with some forensic hand writing analysis expertise.

6. Automatic Text Generation

This is an interesting task, where a corpus of text is learned and from this model new text is generated, word-by-word or character-by-character.

The model is capable of learning how to spell, punctuate, form sentiences and even capture the style of the text in the corpus.

Large recurrent neural networks are used to learn the relationship between items in the sequences of input strings and then generate text. More recently LSTM recurrent neural networks are demonstrating great success on this problem using a character-based model, generating one character at time.

7. Automatic Image Caption Generation

Automatic image captioning is the task where given an image the system must generate a caption that describes the contents of the image.

In 2014, there were an explosion of deep learning algorithms achieving very impressive results on this problem, leveraging the work from top models for object classification and object detection in photographs.

Once you can detect objects in photographs and generate labels for those objects, you can see that the next step is to turn those labels into a coherent sentence description.

This is one of those results that knocked my socks off and still does. Very impressive indeed.

Generally, the systems involve the use of very large convolutional neural networks for the object detection in the photographs and then a recurrent neural network like an LSTM to turn the labels into a coherent sentence.

8. Automatic Game Playing

This is a task where a model learns how to play a computer game based only on the pixels on the screen.

This very difficult task is the domain of deep reinforcement models and is the breakthrough that DeepMind (now part of google) is renown for achieving.

History

The **history of Deep Learning** can be traced back to 1943, when Walter Pitts and Warren McCulloch created a computer model based on the **neural networks** of the human brain. They used a combination of algorithms and mathematics they called “threshold logic” to mimic the thought process.

Deep learning is an increasingly popular subset of **machine learning**. **Deep learning models** are built using **neural networks**. A **neural network** takes in inputs, which are then processed in hidden layers using weights that are adjusted during training. ... Keras is a user-friendly **neural network** library written in Python.

Question – What is semi – supervised learning ?

Answer - **Semi-supervised learning** is an approach to machine learning that combines a small amount of labeled data with a large amount of unlabeled data during training. Semi-supervised learning falls between unsupervised learning (with no labeled training data) and supervised learning (with only labeled training data).

Unlabeled data, when used in conjunction with a small amount of labeled data, can produce considerable improvement in learning accuracy. The acquisition of labeled data for a learning problem often requires a skilled human agent (e.g. to transcribe an audio segment) or a physical experiment (e.g. determining the 3D structure of a protein or determining whether there is oil at a particular location). The cost associated with the labeling process thus may render large, fully labeled training sets infeasible, whereas acquisition of unlabeled data is relatively inexpensive. In such situations, semi-supervised learning can be of great practical value. Semi-supervised learning is also of theoretical interest in machine learning and as a model for human learning.

Today's Machine Learning algorithms can be broadly classified into three categories, Supervised Learning, Unsupervised Learning and Reinforcement Learning. Casting Reinforced Learning aside, the primary two categories of Machine Learning problems are Supervised and Unsupervised Learning. The basic difference between the two is that Supervised Learning datasets have an output label associated with each tuple while Unsupervised Learning datasets do not.

The most basic disadvantage of any **Supervised Learning** algorithm is that the dataset has to be hand-labeled either by a Machine Learning Engineer or a Data Scientist. This is a very *costly process*, especially when dealing with large volumes of data. The most basic disadvantage of any **Unsupervised Learning** is that it's **application spectrum is limited**.

To counter these disadvantages, the concept of **Semi-Supervised Learning** was introduced. In this type of learning, the algorithm is trained upon a combination of labeled and unlabeled data. Typically, this combination will contain a very small amount of labeled data and a very large amount of unlabeled data. The basic procedure involved is that first, the programmer will cluster similar data using an unsupervised learning algorithm and then use the existing labeled data to label the rest of the unlabeled data. The typical use cases of such type of algorithm have a common property among them – The acquisition of unlabeled data is relatively cheap while labeling the said data is very expensive.

Intuitively, one may imagine the three types of learning algorithms as Supervised learning where a student is under the supervision of a teacher at both home and school, Unsupervised learning where a student has to figure out a concept himself and Semi-Supervised learning where a teacher teaches a few concepts in class and gives questions as homework which are based on similar concepts.

A Semi-Supervised algorithm assumes the following about the data –

1. **Continuity Assumption:** The algorithm assumes that the points which are closer to each other are more likely to have the same output label.
2. **Cluster Assumption:** The data can be divided into discrete clusters and points in the same cluster are more likely to share an output label.
3. **Manifold Assumption:** The data lie approximately on a manifold of much lower dimension than the input space. This assumption allows the use of distances and densities which are defined on a manifold.

Practical applications of Semi-Supervised Learning –

1. **Speech Analysis:** Since labeling of audio files is a very intensive task, Semi-Supervised learning is a very natural approach to solve this problem.
2. **Internet Content Classification:** Labeling each webpage is an impractical and unfeasible process and thus uses Semi-Supervised learning algorithms. Even the Google search algorithm uses a variant of Semi-Supervised learning to rank the relevance of a webpage for a given query.
3. **Protein Sequence Classification:** Since DNA strands are typically very large in size, the rise of Semi-Supervised learning has been imminent in this field.

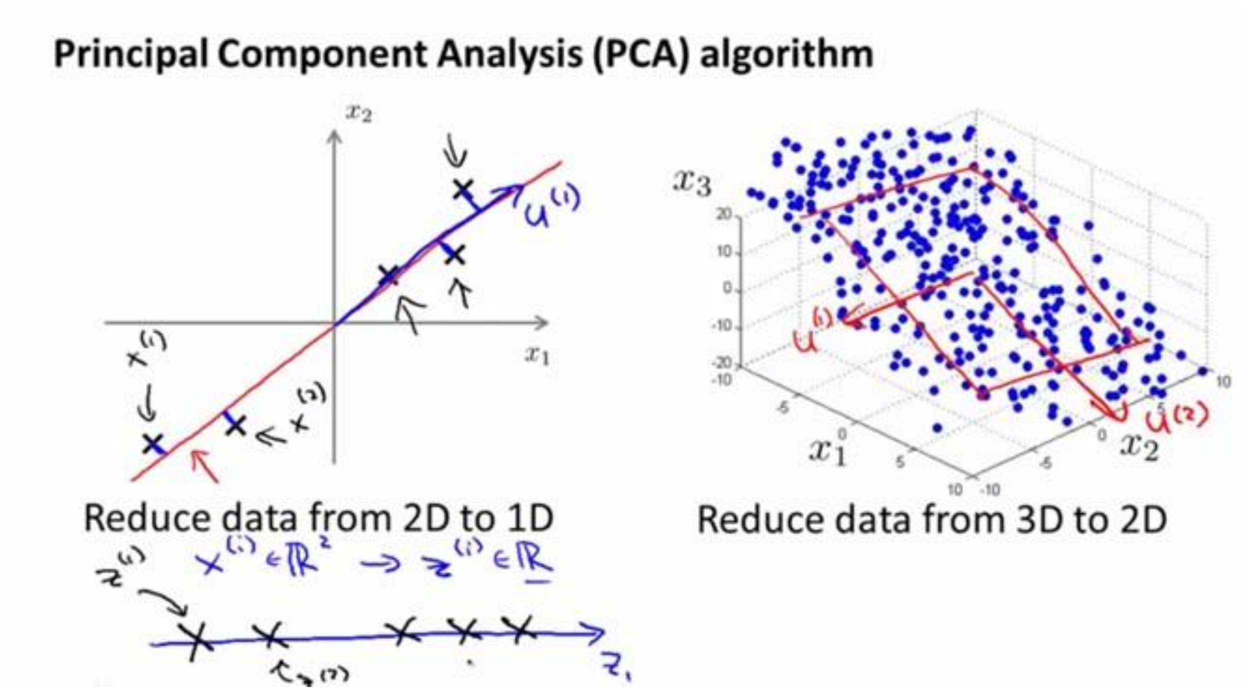
Google, in 2016 launched a new Semi-Supervised learning tool called Google Expander and you can learn more about it

Question - What is PCA (Principle Component Analysis) and RNN .

Answer - The main idea of principal component analysis (PCA) is to reduce the dimensionality of a data set consisting of many variables correlated with each other, either heavily or lightly, while retaining the variation present in the dataset, up to the maximum extent. The same is done by transforming the variables to a new set of variables, which are known as the principal components (or simply, the PCs) and are orthogonal, ordered such that the retention of variation present in the original variables decreases as we move down in the order. So, in this way, the 1st principal component retains maximum variation that was present in the original components. The principal components are the eigenvectors of a covariance matrix, and hence they are orthogonal.

Importantly, the dataset on which PCA technique is to be used must be scaled. The results are also sensitive to the relative scaling. As a layman, it is a method of summarizing data. Imagine some wine bottles on a dining table. Each wine is described by its attributes like colour, strength, age, etc. But redundancy will arise because many of them will measure related properties. So what PCA will do in this case is summarize each wine in the stock with less characteristics.

Intuitively, Principal Component Analysis can supply the user with a lower-dimensional picture, a projection or "shadow" of this object when viewed from its most informative viewpoint.



- Dimensionality : It is the number of random variables in a dataset or simply the number of features, or rather more simply, the number of columns present in your dataset.
- Correlation : It shows how strongly two variable are related to each other. The value of the same ranges for -1 to +1. Positive indicates that when one variable increases, the other increases as well, while negative indicates the other decreases on increasing the former. And the modulus value of indicates the strength of relation.
- Orthogonal: Uncorrelated to each other, i.e., correlation between any pair of variables is 0.
- Eigenvectors:Eigenvectors and Eigenvalues are in itself a big domain, let's restrict ourselves to the knowledge of the same which we would require here. So, consider a non-zero vector ***v***. It is an eigenvector of a square matrix ***A***, if ***Av*** is a scalar multiple of **v**. Or simply:

$$Av = \lambda v$$

Here, ***v*** is the eigenvector and λ is the eigenvalue associated with it.

- Covariance Matrix: This matrix consists of the covariances between the pairs of variables. The *(i,j)*th element is the covariance between *i*-th and *j*-th variable.
-
- **Implementing PCA on a 2-D Dataset**
- **Step 1: Normalize the data**
- First step is to normalize the data that we have so that PCA works properly. This is done by subtracting the respective means from the numbers in the respective column. So if we have two dimensions X and Y, all X become x- and all Y become y-. This produces a dataset whose mean is zero.
- **Step 2: Calculate the covariance matrix**
- Since the dataset we took is 2-dimensional, this will result in a 2x2 Covariance matrix.

$$Matrix(Covariance) = \begin{bmatrix} Var[X_1] & Cov[X_1, X_2] \\ Cov[X_2, X_1] & Var[X_2] \end{bmatrix}$$

-
- Please note that $Var[X_1] = Cov[X_1,X_1]$ and $Var[X_2] = Cov[X_2,X_2]$.
- **Step 3: Calculate the eigenvalues and eigenvectors**

- Next step is to calculate the eigenvalues and eigenvectors for the covariance matrix. The same is possible because it is a square matrix. λ is an eigenvalue for a matrix A if it is a solution of the characteristic equation:
- $det(\lambda I - A) = 0$
- Where, I is the identity matrix of the same dimension as A which is a required condition for the matrix subtraction as well in this case and '*det*' is the determinant of the matrix. For each eigenvalue λ, a corresponding eigen-vector v, can be found by solving:
- $(\lambda I - A)v = 0$
- **Step 4: Choosing components and forming a feature vector:**
- We order the eigenvalues from largest to smallest so that it gives us the components in order or significance. Here comes the dimensionality reduction part. If we have a dataset with n variables, then we have the corresponding n eigenvalues and eigenvectors. It turns out that the eigenvector corresponding to the highest eigenvalue is the principal component of the dataset and it is our call as to how many eigenvalues we choose to proceed our analysis with. To reduce the dimensions, we choose the first p eigenvalues and ignore the rest. We do lose out some information in the process, but if the eigenvalues are small, we do not lose much.
- Next we form a feature vector which is a matrix of vectors, in our case, the eigenvectors. In fact, only those eigenvectors which we want to proceed with. Since we just have 2 dimensions in the running example, we can either choose the one corresponding to the greater eigenvalue or simply take both.
-
- $Feature\ Vector = (eig_1, eig_2)$
-
- **Step 5: Forming Principal Components:**
- This is the final step where we actually form the principal components using all the math we did till here. For the same, we take the transpose of the feature vector and left-multiply it with the transpose of scaled version of original dataset.
- $NewData = FeatureVector^T \times ScaledData^T$
- Here,
- *NewData* is the Matrix consisting of the principal components,
- *FeatureVector* is the matrix we formed using the eigenvectors we chose to keep, and

- *ScaledData* is the scaled version of original dataset
- ('T' in the superscript denotes transpose of a matrix which is formed by interchanging the rows to columns and vice versa. In particular, a 2x3 matrix has a transpose of size 3x2)
-
- If we go back to the theory of eigenvalues and eigenvectors, we see that, essentially, eigenvectors provide us with information about the patterns in the data. In particular, in the running example of 2-D set, if we plot the eigenvectors on the scatterplot of data, we find that the principal eigenvector (corresponding to the largest eigenvalue) actually fits well with the data. The other one, being perpendicular to it, does not carry much information and hence, we are at not much loss when deprecating it, hence reducing the dimension.
- All the eigenvectors of a matrix are perpendicular to each other. So, in PCA, what we do is represent or transform the original dataset using these orthogonal (perpendicular) eigenvectors instead of representing on normal x and y axes. We have now classified our data points as a combination of contributions from both x and y. The difference lies when we actually disregard one or many eigenvectors, hence, reducing the dimension of the dataset. Otherwise, in case, we take all the eigenvectors in account, we are just transforming the co-ordinates and hence, not serving the purpose.
-

Applications of Principal Component Analysis

- PCA is predominantly used as a dimensionality reduction technique in domains like facial recognition, computer vision and image compression. It is also used for finding patterns in data of high dimension in the field of finance, data mining, bioinformatics, psychology, etc.

Pre-solved code recipes usually help in finishing your projects faster.

- **PCA for images:**
- You must be wondering many a times show can a machine read images or do some calculations using just images and no numbers. We will try to answer a part of that now. For simplicity, we will be restricting our discussion to square images only. Any square image of size NxN pixels can be represented as a NxN matrix where each element is the intensity

value of the image. (The image is formed placing the rows of pixels one after the other to form one single image.) So if you have a set of images, we can form a matrix out of these matrices, considering a row of pixels as a vector, we are ready to start principal component analysis on it. How is it useful ?

- Say you are given an image to recognize which is not a part of the previous set. The machine checks the differences between the to-be-recognized image and each of the principal components. It turns out that the process performs well if PCA is applied and the differences are taken from the 'transformed' matrix. Also, applying PCA gives us the liberty to leave out some of the components without losing out much information and thus reducing the complexity of the problem.
- For image compression, on taking out less significant eigenvectors, we can actually decrease the size of the image for storage. But to mention, on reproducing the original image from this will lose out some information for obvious reasons.

Usage in programming:

- For application of PCA, you can hard-code the whole process in any programming language, be it C++, R, Python, etc. or directly use the libraries made available by contributors. However, it is recommended to hard-code in case the problem is not too complex so that you actually get to see what exactly is happening in the back-end when the analysis is being done and also understand the corner cases. Just for instance, in R, there are libraries called princomp, HSAUR, prcomp, etc. which can be used for direct application.

RNN

A recurrent neural network (RNN) is a type of artificial neural network commonly used in speech recognition and natural language processing (NLP). RNNs are designed to recognize a data's sequential characteristics and use patterns to predict the next likely scenario.

RNNs are used in deep learning and in the development of models that simulate the activity of neurons in the human brain. They are especially powerful in use cases in which context is critical to predicting an outcome and are distinct from other types of artificial neural networks because they use feedback loops to process a sequence of data that informs the final output, which can also be a sequence of data . These feedback loops allow information to persist; the effect is often described as memory.

RNN use cases tend to be connected to language models in which knowing the next letter in a word or the next word in a sentence is predicated on the data that comes before it. A compelling experiment involves an RNN trained with the works of Shakespeare to produce Shakespeare-like prose -- successfully. Writing by RNNs is a form of computational creativity. This simulation of human creativity is made possible by the AI's understanding of grammar and semantics learned from its training set.

Question - Explain Back propagation with its algorithm.

Answer - **Backpropagation** is the central mechanism by which **neural networks learn**. It is the messenger telling the network whether or not the net made a mistake when it made a prediction. ... Forward propagation is when a data instance sends its signal through a network's parameters toward the prediction at the end.

Back-propagation is the essence of neural net training. It is the method of fine-tuning the weights of a neural net based on the error rate obtained in the previous epoch (i.e., iteration). Proper tuning of the weights allows you to reduce error rates and to make the model reliable by increasing its generalization.

Backpropagation is a short form for "backward propagation of errors." It is a standard method of training artificial neural networks. This method helps to calculate the gradient of a loss function with respects to all the weights in the network.

Types of Backpropagation Networks

Two Types of Backpropagation Networks are:

- Static Back-propagation

- Recurrent Backpropagation

Static back-propagation:

It is one kind of backpropagation network which produces a mapping of a static input for static output. It is useful to solve static classification issues like optical character recognition.

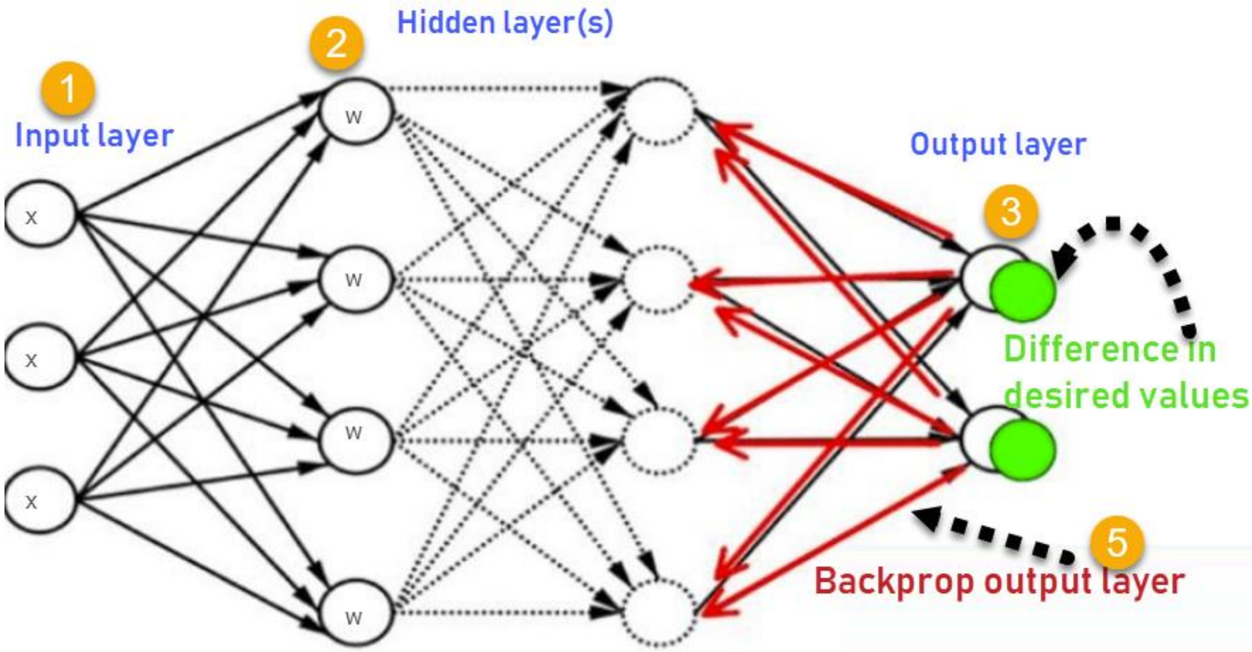
Recurrent Backpropagation:

Recurrent backpropagation is fed forward until a fixed value is achieved. After that, the error is computed and propagated backward.

The main difference between both of these methods is: that the mapping is rapid in static back-propagation while it is nonstatic in recurrent backpropagation.

Algorithm

Consider the following diagram



1. Inputs X, arrive through the preconnected path
2. Input is modeled using real weights W. The weights are usually randomly selected.
3. Calculate the output for every neuron from the input layer, to the hidden layers, to the output layer.
4. Calculate the error in the outputs

```
ErrorB= Actual Output – Desired Output
```

5. Travel back from the output layer to the hidden layer to adjust the weights such that the error is decreased.

Why We Need Backpropagation?

Most prominent advantages of Backpropagation are:

- Backpropagation is fast, simple and easy to program
- It has no parameters to tune apart from the numbers of input
- It is a flexible method as it does not require prior knowledge about the network
- It is a standard method that generally works well
- It does not need any special mention of the features of the function to be learned.

Disadvantages of using Backpropagation

- The actual performance of backpropagation on a specific problem is dependent on the input data.
- Backpropagation can be quite sensitive to noisy data
- You need to use the matrix-based approach for backpropagation instead of mini-batch.

Question - Short notes :-

A : - Deep Reinforcement Learning

Answer -**Deep reinforcement learning** combines artificial neural networks with a **reinforcement learning** architecture that enables software-defined agents to **learn** the best actions possible in virtual environment in order to attain their goals.

Deep reinforcement learning is the combination of reinforcement learning (RL) and deep learning. This field of research has been able to solve a wide range of complex decision-making tasks that were previously out of reach for a machine. Thus, deep RL opens up many new applications in domains such as healthcare, robotics, smart grids, finance, and many more. This manuscript provides an introduction to deep reinforcement learning models, algorithms and techniques. Particular focus is on the aspects related to generalization and how deep RL can be used for practical applications. We assume the reader is familiar with basic machine learning concepts.

B : - Autoencoder Architecture

Answer :-An **autoencoder** is a neural network **architecture** capable of discovering **structure** within data in order to develop a compressed representation of the input. ... Because **autoencoders** learn how to compress the data based on attributes.Autoencoders are an unsupervised learning technique in which we leverage neural networks for the task of **representation learning**. Specifically, we'll design a neural network architecture such that we *impose a bottleneck in the network which forces a* ***compressed*** *knowledge representation of the original input*. If the input features were each independent of one another, this compression and subsequent reconstruction would be a very difficult task. However, if some sort of structure exists in the data (ie. correlations between input features), this structure can be learned and consequently leveraged when forcing the input through the network's bottleneck.

C : - Visual Geometry Group (VGG)

Answer : - **Visual Geometry Group** (VGG) has smaller filters than AlexNet, where each filter is of size 3 x 3 but with a lower stride of one, which effectively captures the same receptive field as a 7 x 7 filter with four strides. It has typically 16-19 layers depending on the particular VGG configuration.

VGG means Visual Geometry Group at University of Oxford. The Convolutional neural networks they developed for winning the ImageNet Challenge 2014 in localization and classification tasks are known as **VGG** nets.

VGG is a convolutional neural network **model** proposed by K. ...Zisserman from the University of Oxford in the paper “Very Deep Convolutional Networks for Large-Scale Image Recognition” . The **model** achieves 92.7% top-5 test accuracy in ImageNet , which is a dataset of over 14 million images belonging to 1000 classes.

D : - Face Recognition Application

Answer : - Facial recognition is being used in many businesses

You’re used to unlocking your door with a key, but maybe not with your face. As strange as it sounds, our physical appearances can now verify payments, grant access and improve existing security systems. Protecting physical and digital possessions is a universal concern which benefits everyone, unless you're a cybercriminal or a kleptomaniac of course. Facial biometrics are gradually being applied to more industries, disrupting design, manufacturing, construction, law enforcement and healthcare. How is facial recognition software affecting these different sectors, and who are the companies and organisations behind its development?

1. Payments

It doesn't take a genius to work out why businesses want payments to be easy. Online shopping and contactless cards are just two examples that demonstrate the seamlessness of postmodern purchases. With FaceTech, however, customers wouldn't even need their cards. In 2016, MasterCard launched a new selfie pay app called MasterCard Identity Check. Customers open the app to confirm a payment using their camera, and that's that. Facial recognition is already used in store and at ATMs, but the next step is to do the same for online payments. Chinese ecommerce firm Alibaba and affiliate payment software Alipay are planning to apply the software to purchases made over the Internet.

2. Access and security

As well as verifying a payment, facial biometrics can be integrated with physical devices and objects. Instead of using passcodes, mobile phones and other consumer electronics will be accessed via owners' facial features. Apple, Samsung and Xiaomi Corp.have all installed FaceTech in their phones. This is only a small scale example, though. In future, it looks like consumers will be able to get into their cars, houses, and other secure physical locations simply by looking at them. Jaguar is already working on walking gait ID – a potential parallel to facial recognition technology. Other corporations are likely to take advantage of this, too. Innovative facial security could be especially useful for a company or organisation that handles sensitive data and needs to keep tight controls on who enters their facilities.

3. Criminal identification

If FaceTech can be used to keep unauthorised people out of facilities, surely it can be used to help put them firmly inside them. This is exactly what the US Federal Bureau of Investigation is attempting to do by using a machine learning algorithm to identify suspects from their driver's licences. The FBI currently have a database which includes half of the national population's faces. This is as useful as it is creepy, giving law enforcers another way of tracking criminals across the country. AI equipped cameras have also been trialled in the UK to identify those smuggling contraband into prisons.

4. Advertising

The ability to collect and collate masses of personal data has given marketers and advertisers the chance to get closer than ever to their target markets. FaceTech could do much the same, by allowing companies to recognise certain demographics – for instance, if the customer is a male between the ages of 12 and 21, the screen might show an ad for the latest FIFA game. Grocery giant Tesco plans to install OptimEyes screens at 450 petrol stations in the UK to deliver targeted ads to customers. According to company CEO Simon Sugar, the cameras could change the face of British retail. Perhaps he's right – but only if the cameras can correctly identify customers. Being classified as the wrong age or gender is far less amusing than having your name spelt wrong on a Starbucks cup.

5. Healthcare

Instead of recognising an individual via FaceTech, medical professionals could identify illnesses by looking at a patient's features. This would alleviate the ongoing strain on medical centres by slashing waiting lists and streamlining the appointment process. The question is, would you really want to find out you had a serious illness from a screen? If it's a choice between a virtual consultation or a month long wait for an appointment, then maybe so. Another application of facial biometrics within healthcare is to secure patient data by using a unique patient photo instead of passwords and usernames.

With a predicted worth of $15 billion by 2025, biometrics is an industry worth watching. It's clear that facial biometrics are a helpful tool for finance, law enforcement, advertising and healthcare, as well as a solution to hacking and identity theft. Of course, FaceTech is by no means foolproof. Gaining access to possessions using physical traits could even be counterintuitive for security. A face, as social robots like Nadine have shown us, is easily replicated. And when it comes to public adoption, some people are reluctant to switch to contactless cards, let alone abandon them completely. For the most part, though, facial recognition technology seems to be encouraging a more seamless relationship between people, payments and possessions.