Course Title: Applied Machine Learning

Course Code: CSE4008

Assignment-1

Faculty Name: Dr. Virendra Singh Kushwah

Name : Prachi Bhatt

Reg No. : 19BAI10111

1. Discuss various applications of supervised and unsupervised learning and also justify which type of learning is better.

Ans: Supervised learning as the name indicates the presence of a supervisor as a teacher. Basically supervised learning is a learning in which we teach or train the machine using data which is well labeled that means some data is already tagged with the correct answer. After that, the machine is provided with a new set of examples(data) so that supervised learning algorithm analyses the training data(set of training examples) and produces a correct outcome from labeled data.

**Applications of Supervised Learning**

Supervised Learning Algorithms are used in a variety of applications. Let’s go through some of the most well-known applications.

* **BioInformatics** – This is one of the most well-known applications of Supervised Learning because most of us use it in our day-to-day lives. BioInformatics is the storage of Biological Information of us humans such as fingerprints, iris texture, earlobe and so on. Cellphones of today are capable of learning our biological information and are then able to authenticate us bringing up the security of the system. Smartphones such as iPhones, Google Pixel are capable of facial recognition while OnePlus, Samsung is capable of In-display finger recognition.
* [**Speech Recognition**](https://www.edureka.co/blog/speech-recognition-python/) – This is the kind of application where you teach the algorithm about your voice and it will be able to recognize you. The most well-known real-world applications are virtual assistants such as Google Assistant and Siri, which will wake up to the keyword with your voice only.
* **Spam Detection** – This application is used where the unreal or computer-based messages and E-Mails are to be blocked. G-Mail has an algorithm that learns the different keywords which could be fake such as “You are the winner of something” and so forth and blocks those messages directly. OnePlus Messages App gives the user the task of making the application learn which keywords need to be blocked and the app will block those messages with the keyword.
* [**Object-Recognition**](https://www.edureka.co/blog/tensorflow-object-detection-tutorial/)**for Vision** – This kind of application is used when you need to identify something. You have a huge dataset which you use to teach your algorithm and this can be used to recognize a new instance. [Raspberry Pi](https://www.edureka.co/blog/raspberry-pi-tutorial/) algorithms which detect objects are the most well-known example

Unsupervised learning is the training of machine using information that is neither classified nor labeled and allowing the algorithm to act on that information without guidance. Here the task of machine is to group unsorted information according to similarities, patterns and differences without any prior training of data.

**Applications of Unsupervised Learning**

Unsupervised Learning helps in a variety of ways which can be used to solve various real-world problems.

* They help us in understanding patterns which can be used to cluster the data points based on various features.
* Understanding various defects in the dataset which we would not be able to detect initially.
* They help in mapping the various items based on the dependencies of each other.
* Cleansing the datasets by removing features which are not really required for the machine to learn from.

This ultimately leads to applications which are helpful to us. Certain examples of where Unsupervised Learning algorithms are used are discussed below:

* **AirBnB** – This is a great application which helps host stays and experiences connecting people all over the world. This application uses Unsupervised Learning where the user queries his or her requirements and Airbnb learns these patterns and recommends stays and experiences which fall under the same group or cluster.
* **Amazon** – Amazon also uses unsupervised learning to learn the customer’s purchase and recommend the products which are most frequently bought together which is an example of association rule mining.
* **Credit-Card Fraud Detection** – Unsupervised Learning algorithms learn about various patterns of the user and their usage of the credit card. If the card is used in parts that do not match the behaviour, an alarm is generated which could possibly be marked fraud and calls are given to you to confirm whether it was you using the card or not.

Supervised learning is a simpler method. Unsupervised learning is computationally complex. Supervised is Highly accurate and trustworthy method.while UL is Less accurate and trustworthy method.

SL Learning method takes place offline.

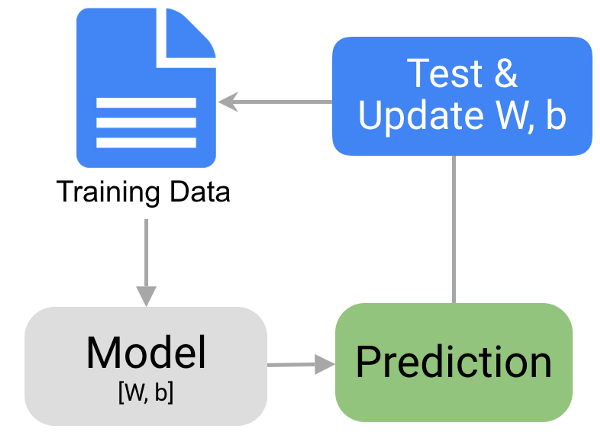
UL Learning method takes place in real time.

**Supervised learning** model produces an accurate result. **Unsupervised learning** model may give less accurate result as compared to **supervised learning**. **Supervised learning** is not close to true Artificial intelligence as in this, we first train the model for each data, and then only it can predict the correct output.

Unlike supervised learning, no teacher is provided that means no training will be given to the machine. Therefore machine is restricted to find the hidden structure in unlabeled data by our-self.

1. Draw seven steps diagram of machine learning process and describe which process is more important to run a model in smooth way. Justify your answer with an example.

### Ans : The 7 Steps of Machine Learning

  
[Image source](https://towardsdatascience.com/the-7-steps-of-machine-learning-2877d7e5548e)

**1 - Data Collection**

* The quantity & quality of your data dictate how accurate our model is
* The outcome of this step is generally a representation of data (Guo simplifies to specifying a table) which we will use for training
* Using pre-collected data, by way of datasets from Kaggle, UCI, etc., still fits into this step

**2 - Data Preparation**

* Wrangle data and prepare it for training
* Clean that which may require it (remove duplicates, correct errors, deal with missing values, normalization, data type conversions, etc.)
* Randomize data, which erases the effects of the particular order in which we collected and/or otherwise prepared our data
* Visualize data to help detect relevant relationships between variables or class imbalances (bias alert!), or perform other exploratory analysis
* Split into training and evaluation sets

**3 - Choose a Model**

* Different algorithms are for different tasks; choose the right one

**4 - Train the Model**

* The goal of training is to answer a question or make a prediction correctly as often as possible
* Linear regression example: algorithm would need to learn values for m (or W) and b (x is input, y is output)
* Each iteration of process is a training step

**5 - Evaluate the Model**

* Uses some metric or combination of metrics to "measure" objective performance of model
* Test the model against previously unseen data
* This unseen data is meant to be somewhat representative of model performance in the real world, but still helps tune the model (as opposed to test data, which does not)
* Good train/eval split? 80/20, 70/30, or similar, depending on domain, data availability, dataset particulars, etc.

**6 - Parameter Tuning**

* This step refers to hyperparameter tuning, which is an "artform" as opposed to a science
* Tune model parameters for improved performance
* Simple model hyperparameters may include: number of training steps, learning rate, initialization values and distribution, etc.

**7 - Make Predictions**

* Using further (test set) data which have, until this point, been withheld from the model (and for which class labels are known), are used to test the model; a better approximation of how the model will perform in the real world

The step of gathering data is the foundation of the machine learning process. Mistakes such as

choosing the incorrect features or focusing on limited types of entries for the data set may render

the model completely ineffective. That is why it is imperative that the necessary considerations are

made when gathering data as the errors made in this stage would only amplify as we progress to

latter stages.

For the purpose of developing our machine learning model, our first step would be to gather

relevant data that can be used to differentiate between the 2 fruits. Different parameters can be

used to classify a fruit as either an orange or apple. For the sake of simplicity, we would only take

2 features that our model would utilize in order to perform its operation. The first feature would

be the color of the fruit itself and the second one being the shape of the fruit. Using these features,

we would hope that our model can accurately differentiate between the 2 fruits.

Color Shape Apple or Orange?

Red Round Conical Apple

Orange Round Orange

A mechanism would be needed to gather the data for our 2 chosen features. For instance, for

collecting data on color, we may use a spectrometer and, for the shape data, we may use pictures

of the fruits so that they can be treated as 2D figures. For the purpose of collecting data, we would

try to get as many different types of apples and orange as possible in order to create diverse data

sets for our features. For this purpose, we may try to search the markets for oranges and apples

that may be from different parts of the world.

3. “To increase degrees of a polynomial regression equation improves better model in machine learning.”If yes, justify your answer by a diagram.

4. What do you mean by Information Gain and Entropy? How is it used to build the Decision tree in algorithm? Illustrate using an example.

Ans:

**Decision Tree :**Decision tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

**1. Information Gain**  
When we use a node in a decision tree to partition the training instances into smaller subsets the entropy changes. Information gain is a measure of this change in entropy.  
***Definition***: Suppose S is a set of instances, A is an attribute, Sv is the subset of S with A = v, and Values (A) is the set of all possible values of A, then

**Entropy**  
Entropy is the measure of uncertainty of a random variable, it characterizes the impurity of an arbitrary collection of examples. The higher the entropy more the information content.  
***Definition***: Suppose S is a set of instances, A is an attribute, Sv is the subset of S with A = v, and Values (A) is the set of all possible values of A, then  
  
Example:

For the set X = {a,a,a,b,b,b,b,b}

Total intances: 8

Instances of b: 5

Instances of a: 3

= -[0.375 \* (-1.415) + 0.625 \* (-0.678)]

=-(-0.53-0.424)

= 0.954

**Building Decision Tree using Information Gain**  
**The essentials:**

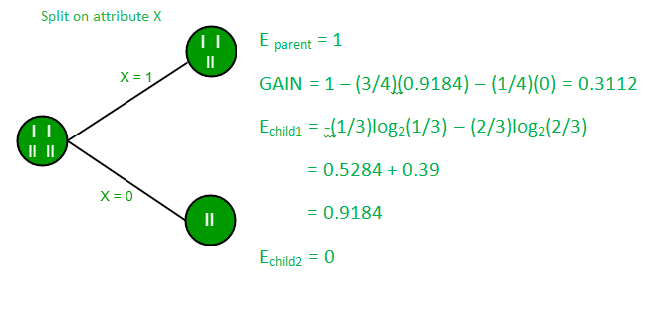
* Start with all training instances associated with the root node
* Use info gain to choose which attribute to label each node with
* *Note:* No root-to-leaf path should contain the same discrete attribute twice
* Recursively construct each subtree on the subset of training instances that would be classified down that path in the tree.

**The border cases:**

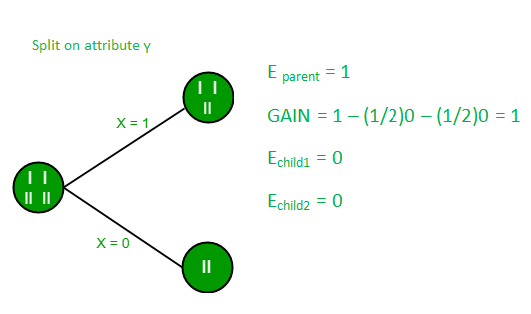
* If all positive or all negative training instances remain, label that node “yes” or “no” accordingly
* If no attributes remain, label with a majority vote of training instances left at that node
* If no instances remain, label with a majority vote of the parent’s training instances

**Example:**  
Now, lets draw a Decision Tree for the following data using Information gain.

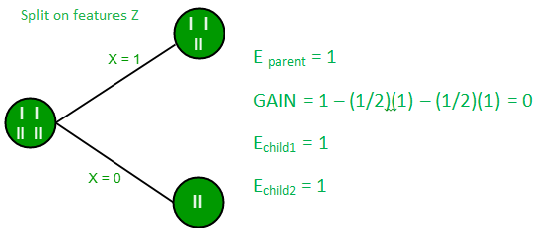
|  |  |  |  |
| --- | --- | --- | --- |
| **Training set: 3 features and 2 classes** |  |  |  |
|  |  |  |  |
|  |  |  |  |

Here, we have 3 features and 2 output classes.  
To build a decision tree using Information gain. We will take each of the feature and calculate the information for each feature.  


**Split on feature X**

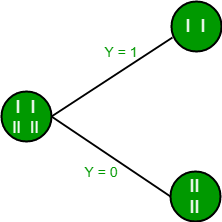


**Split on feature Y**



**Split on feature Z**

From the above images we can see that the information gain is maximum when we make a split on feature Y. So, for the root node best suited feature is feature Y. Now we can see that while splitting the dataset by feature Y, the child contains pure subset of the target variable. So we don’t need to further split the dataset.

The final tree for the above dataset would be look like this:  


5.Explain why decision tree classifier is not better for classification problem?

6.Compare and Contrast Logistic Regression and Support Vector Machine for any classification problem.

Hence, key points are:

* SVM try to maximize the margin between the closest support vectors whereas logistic regression maximize the posterior class probability
* SVM is deterministic (but we can use Platts model for probability score) while LR is probabilistic.
* For the kernel space, SVM is faster

| S.No. | Logistic Regression | Support Vector Machine |
| --- | --- | --- |
| 1. | It is an algorithm used for solving classification problems. | It is a model used for both classification and regression. |
| 2. | It is not used to find the best margin, instead, it can have different decision boundaries with different weights that are near the optimal point. | it tries to find the “best” margin (distance between the line and the support vectors) that separates the classes and thus reduces the risk of error on the data. |
| 3. | It works with already identified identified independent variable. | It works well with unstructured and semi-structured data like text and images. |
| 4. | It is based on statistical approach. | It is based on geometrical properties of the data. |
| 5. | It is vulnerable to overfitting. | The risk of overfitting is less in SVM. |
| 6. | Problems to apply logistic regression algorithm.  1. Cancer Detection: It can be used to detect if a patient has cancer(1) or not(0)    2. Test Score: Predict if the student is passed(1) or not(0).  3. Marketing: Predict if a customer will purchase a product(1) or not(0). |  |

Problems that can be solved using SVM

1. Image Classification

2. Recognizing handwriting

3. Cancer Detection

7.Calculate accuracy, recall and precision values from the following confusion matrix.

|  |  |
| --- | --- |
| 50 | 25 |
| 40 | 90 |

8.Write five important real-life applications of neural network and describe any one of them in details.

Ans:

## Online Shopping

### Search

### Recommendations

## Banking/Personal Finance

### (Cheque Deposits Through Mobile)

### Fraud Prevention

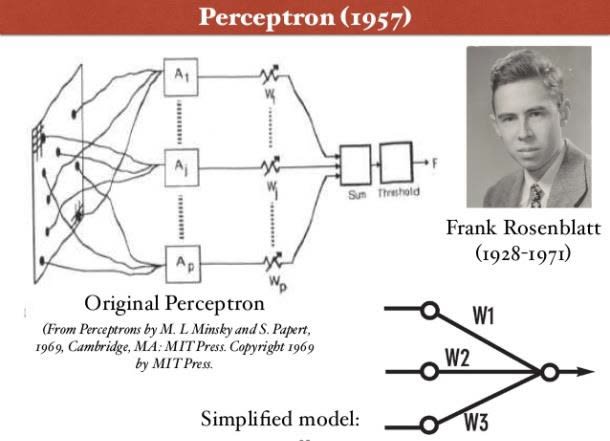
### Recommendations:

Amazon shows you recommendations using its “customers who viewed this item also viewed”,  “customers who bought this item also bought”, and also via curated recommendations on your homepage, on the bottom of the item pages, and through emails. Amazon makes use of Artificial Neural Networks to train its algorithms to learn the pattern and behaviour of its users. This, in turn, helps Amazon provide even better and customized recommendations.

9.Explain the concept of a Perceptron with a neat diagram.

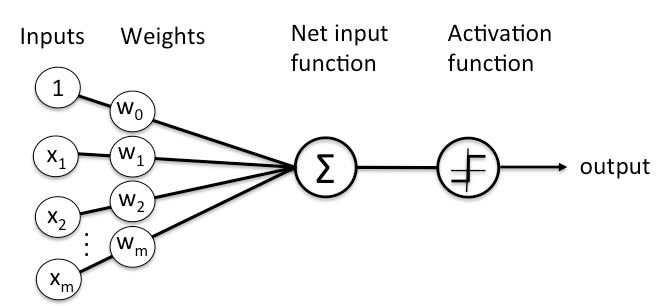
## Perceptron

A perceptron is a neural network unit (an artificial neuron) that does certain computations to detect features or business intelligence in the input data.



Perceptron was introduced by Frank Rosenblatt in 1957. He proposed a Perceptron learning rule based on the original MCP neuron.

A Perceptron is an algorithm for supervised learning of binary classifiers. This algorithm enables neurons to learn and processes elements in the training set one at a time.

There are two types of Perceptrons: Single layer and Multilayer.

Single layer Perceptrons can learn only linearly separable patterns.

Multilayer Perceptrons or feedforward neural networks with two or more layers have the greater processing power.

The Perceptron algorithm learns the weights for the input signals in order to draw a linear decision boundary.

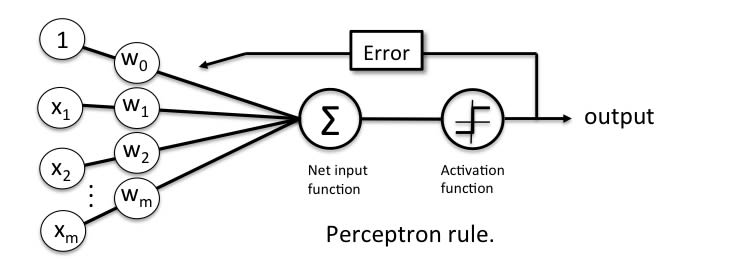
This enables you to distinguish between the two linearly separable classes +1 and -1.

Note: Supervised Learning is a type of Machine Learning used to learn models from labeled training data. It enables output prediction for future or unseen data.

Let us focus on the Perceptron Learning Rule in the next section.

## Perceptron Learning Rule

Perceptron Learning Rule states that the algorithm would automatically learn the optimal weight coefficients. The input features are then multiplied with these weights to determine if a neuron fires or not.

The Perceptron receives multiple input signals, and if the sum of the input signals exceeds a certain threshold, it either outputs a signal or does not return an output. In the context of supervised learning and classification, this can then be used to predict the class of a sample.

In the next section, let us focus on the perceptron function.

## Perceptron Function

Perceptron is a function that maps its input “x,” which is multiplied with the learned weight coefficient; an output value ”f(x)”is generated.

https://www.simplilearn.com/ice9/free_resources_article_thumb/mathematical-function-representation-of-perceptron.jpg

In the equation given above:

“w” = vector of real-valued weights

“b” = bias (an element that adjusts the boundary away from origin without any dependence on the input value)

“x” = vector of input x values

https://www.simplilearn.com/ice9/free_resources_article_thumb/value-of-vector-of-input-x-values-in-perceptron-function.jpg

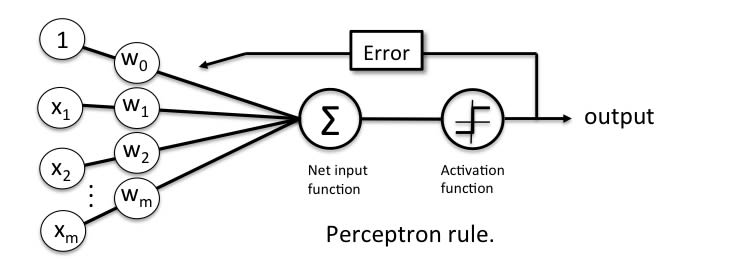
“m” = number of inputs to the Perceptron

The output can be represented as “1” or “0.”  It can also be represented as “1” or “-1” depending on which activation function is used.

Let us learn the inputs of a perceptron in the next section.

## Inputs of a Perceptron

A Perceptron accepts inputs, moderates them with certain weight values, then applies the transformation function to output the final result. The above below shows a Perceptron with a Boolean output.



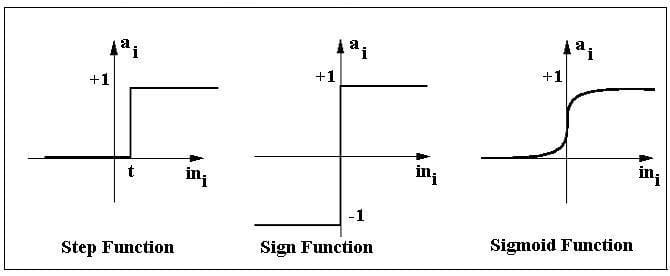
A Boolean output is based on inputs such as salaried, married, age, past credit profile, etc. It has only two values: Yes and No or True and False. The summation function “∑” multiplies all inputs of “x” by weights “w” and then adds them up as follows:

https://www.simplilearn.com/ice9/free_resources_article_thumb/summation-function-in-perceptron-formula-for-output.jpg

In the next section, let us discuss the activation functions of perceptron.

## Activation Functions of Perceptron

The activation function applies a step rule (convert the numerical output into +1 or -1) to check if the output of the weighting function is greater than zero or not.

For example:

If ∑ wixi> 0 => then final output “o” = 1 (issue bank loan)

Else, final output “o” = -1 (deny bank loan)

Step function gets triggered above a certain value of the neuron output; else it outputs zero. Sign Function outputs +1 or -1 depending on whether neuron output is greater than zero or not. Sigmoid is the S-curve and outputs a value between 0 and 1.

## Output of Perceptron

Perceptron with a Boolean output:

Inputs: x1…xn

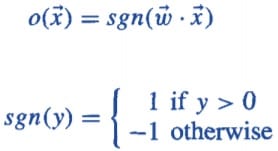
Output: o(x1….xn)

https://www.simplilearn.com/ice9/free_resources_article_thumb/equation-for-perceptron-with-a-boolean-output.jpg

Weights: wi=> contribution of input xi to the Perceptron output;

w0=> bias or threshold

If ∑w.x > 0, output is +1, else -1. The neuron gets triggered only when weighted input reaches a certain threshold value.



An output of +1 specifies that the neuron is triggered. An output of -1 specifies that the neuron did not get triggered.

“sgn” stands for sign function with output +1 or -1.

Want to check the Course Preview of Deep Learing? [Click here to watch!](https://www.simplilearn.com/deep-learning-course-with-tensorflow-training?source=GhPreviewCTAText#/course-preview)

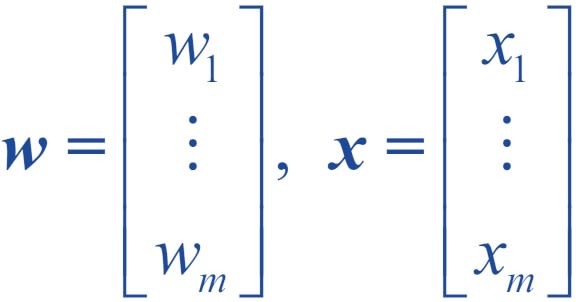
## Error in Perceptron

In the Perceptron Learning Rule, the predicted output is compared with the known output. If it does not match, the error is propagated backward to allow weight adjustment to happen.

Let us discuss the decision function of Perceptron in the next section.

## Perceptron: Decision Function

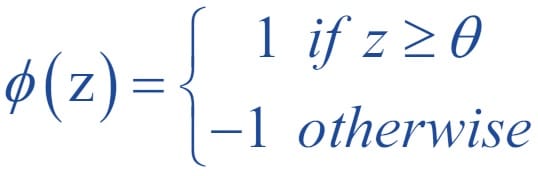
A decision function φ(z) of Perceptron is defined to take a linear combination of x and w vectors.



The value z in the decision function is given by:

https://www.simplilearn.com/ice9/free_resources_article_thumb/value-of-z-in-decision-function.jpg

The decision function is +1 if z is greater than a threshold θ, and it is -1 otherwise.



This is the Perceptron algorithm.

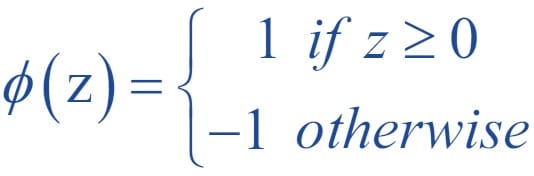
**Bias Unit**

For simplicity, the threshold θ can be brought to the left and represented as w0x0, where w0= -θ and x0= 1.



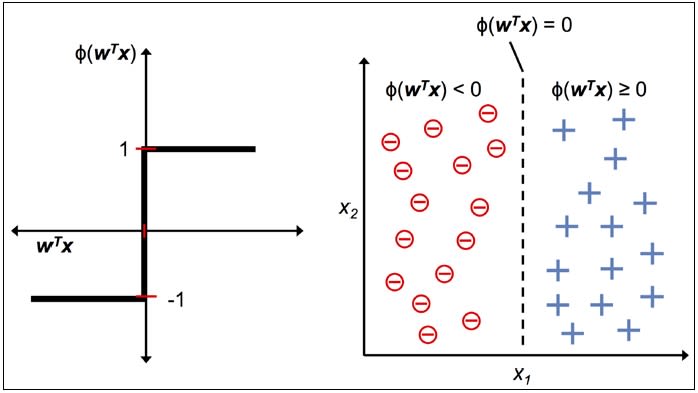
The value w0  is called the bias unit.

The decision function then becomes:



**Output**

The figure shows how the decision function squashes wTx to either +1 or -1 and how it can be used to discriminate between two linearly separable classes.



## Perceptron at a Glance

Perceptron has the following characteristics:

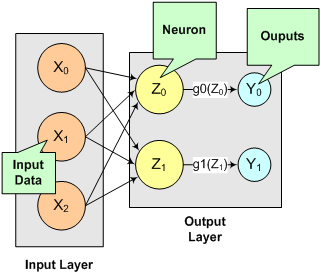
* Perceptron is an algorithm for Supervised Learning of single layer binary linear classifier.
* Optimal weight coefficients are automatically learned.
* Weights are multiplied with the input features and decision is made if the neuron is fired or not.
* Activation function applies a step rule to check if the output of the weighting function is greater than zero.
* Linear decision boundary is drawn enabling the distinction between the two linearly separable classes +1 and -1.
* If the sum of the input signals exceeds a certain threshold, it outputs a signal; otherwise, there is no output.

Types of activation functions include the sign, step, and sigmoid functions.

10.Draw diagram of feed forward multi layers neural network and explain with an example.(ppt best ans)

## Multilayer Feedforward Neural Networks

A multilayer feedforward neural network is an interconnection of perceptrons in which data and calculations flow in a single direction, from the input data to the outputs.  The number of layers in a neural network is the number of layers of perceptrons.  The simplest neural network is one with a single input layer and an output layer of perceptrons.  The network in Figure 13-7 illustrates this type of network.  Technically, this is referred to as a one-layer feedforward network with two outputs because the output layer is the only layer with an activation calculation.



*Figure 13- 7:  A Single-Layer Feedforward Neural Net*

In this single-layer feedforward neural network, the network’s inputs are directly connected to the output layer perceptrons, *Z*1 and *Z*2.

The output perceptrons use activation functions, *g*1 and *g*2, to produce the outputs *Y*1 and *Y*2.

Since

https://help.imsl.com/c/6.0/stat/ImagesExt/image26_28.png ,

https://help.imsl.com/c/6.0/stat/ImagesExt/image26_29.png,

and

https://help.imsl.com/c/6.0/stat/ImagesExt/image26_30.png.

When the activation functions *g*1 and *g*2 are identity activation functions, the single-layer neural net is equivalent to a linear regression model.  Similarly, if *g*1 and *g*2 are logistic activation functions, then the single-layer neural net is equivalent to logistic regression.  Because of this correspondence between single-layer neural networks and linear and logistic regression, single-layer neural networks are rarely used in place of linear and logistic regression.

The next most complicated neural network is one with two layers.  This extra layer is referred to as a hidden layer.  In general there is no restriction on the number of hidden layers.  However, it has been shown mathematically that a two-layer neural network

can accurately reproduce any differentiable function, provided the number of perceptrons in the hidden layer is unlimited.

However, increasing the number of perceptrons increases the number of weights that must be estimated in the network, which in turn increases the execution time for the network.  Instead of increasing the number of perceptrons in the hidden layers to improve accuracy, it is sometimes better to add additional hidden layers, which typically reduce both the total number of network weights and the computational time.  However, in practice, it is uncommon to see neural networks with more than two or three hidden layers.