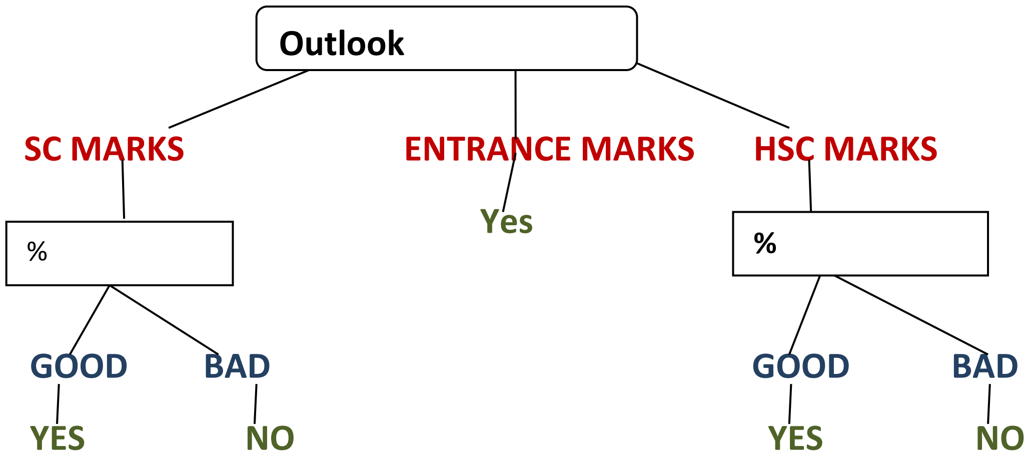
Decision Tree

A decision tree is a flow chart like structure in which each internal node represents a test on a feature (whether a coin flip comes up heads or tails), each leaf node represents a class label and branches represent conjunctions of features that lead to those class labels. The paths from root to leaf represent classification rules.



Above diagram illustrate the basic flow of decision tree for decision making with labels (loan offered (yes)), no loan offered (No).

Decision trees are constructed via an algorithmic approach that identifies ways to split a data set based on different conditions. It is one of the most widely used and practical methods for supervised learning. This algorithm used for both classification and regression tasks.

CART: Classification and Regression trees

Entropy: Entropy is nothing but the measure of disorder.

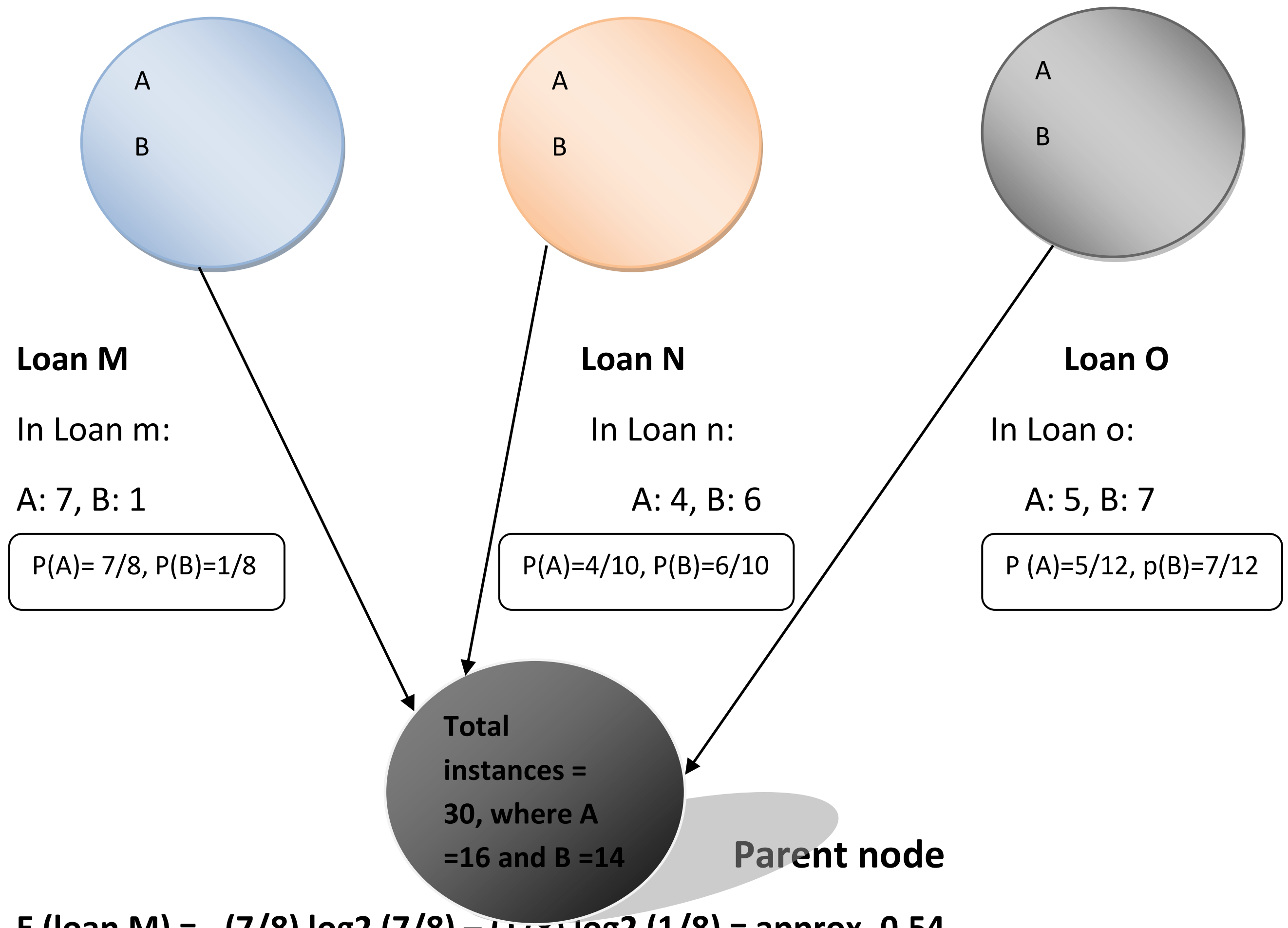
$$E(S) = \sum_{i=1}^{c}(-\text{Pi } \log 2 \text{ Pi})$$

Where 'Pi' is simply the frequentist probability of an element/class 'i' in data. If we have two classes (+ ve and – ve), therefore 'i' here could be either +ve or –ve. As we have 100 data points. Among the 30% belonging from +ve class and 70% belonging from –ve. So, 'P+' would be 3/10 and 'P-'would be 7/10.

- (3/10) * log2 (3/10) – (7/10) * log2 (7/10) which is approximately 0.88(a high level disorder). Entropy is measured between 0 and 1 (depending on the number of classes in the dataset). Entropy can be greater than 1 but it means a high level disorder.

Information gain: Information gain is the reduction in entropy by transforming a dataset and is used in training decision trees. IG is calculated by comparing the entropy of the dataset before and after transformation.

Information gain of y to x (y, x): IG(y, x) = E(y) –E(y/x)



E (loan M) = - (7/8) log2 (7/8) – (1/8) log2 (1/8) = approx. 0.54

E (loan N) = - (4/10) log2 (4/10) – (6/10) log2 (6/10) = approx. 0.97

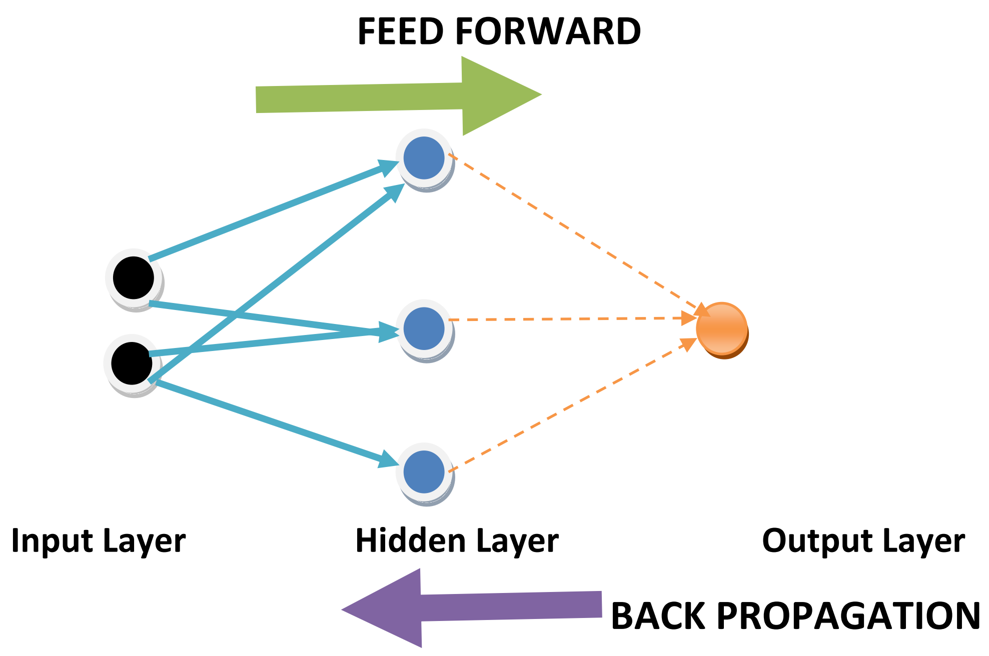
E (loan O) = - (5/12) log2 (5/12) – (7/12) log2 (7/12) = approx. 0.98

Weighted average of entropies of each node:

E (loans) = (8/30) * 0.54 + (10/30) * 0.97 + (12/30) * 0.98 = 0.86

Information gain: E (Parent, loans) = E (Parent) – E (loans) = 1-0.86 = 0.14

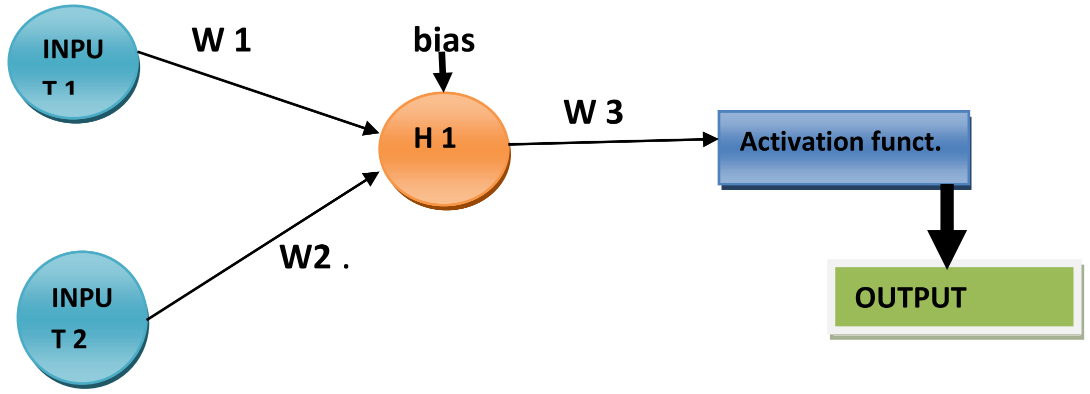
Deep Neural Networks: It is a group of multiple perceptron or multiple neurons at each layer. It is an Artificial Neural Network with multiple layers between the input and output layers. It is a class of machine learning algorithms that uses multiple layers to progressively extract higher level features from the raw input. An ANN is based on a collection of connected units called artificial neurons. Each connection between neurons can transmit a signal and then signal downstream neurons connected to it. There are different types of neural networks but they always consist of the same components: neurons, synapses, weights, biases and functions.



Input is processed in the **forward direction**.

Weights and Biases referred to as W and B, are the learnable parameters of a machine learning model. Weights control the signal between two neurons. In other words, a weight decides how much influence the input will have on the output. Biases are constant and an additional input into the next layer. That will always have the value of 1. Bias units are not influenced by the previous layer but they do have outgoing connections with their own weights.

$$Y = \sum_{n=1}^{\infty}(\mathbf{input} * \mathbf{weight}) + \mathbf{bias}$$

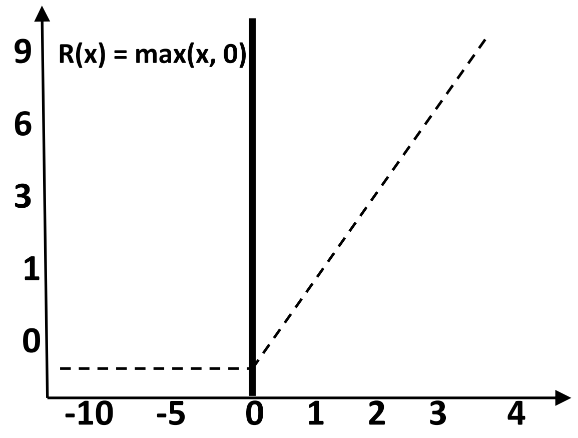


OUTPUT= (input 1*w1+input 2*w2) +bias

Activation Function: An activation function in neural network defines how the weighted sum of the input is transformed into an output from a node or nodes in a layer of the network.

RELU: it stands for rectified linear unit. It is the most widely used activation function. It's implemented in hidden layers of neural networks.

Value range: [0, infinite), Equation: A(x) = max (0, x), the output gives x if x is positive and 0 otherwise. It is faster than sigmoid and tanh function.

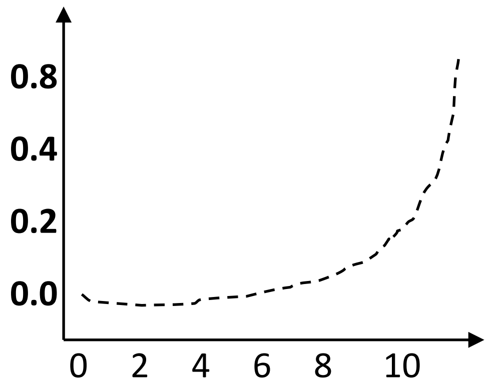


Softmax: the Softmax function is used for when we are trying to handle classification problems. The Softmax function would squeeze the outputs for each class between 0 and 1 and would also divide by the sum of outputs.

Transform values into probabilities. Value is 1 in a list [1, 2].

Probability: p1= exp (1) / (exp (1) + exp (2))

P2= exp (2) / (exp (1) + (exp (2)),,,,,, sum= (p1 +p2)



1. **Input layer:** this layer accepts input features. It provides information from the outside world to the network. In other words, it is a layer whereas we can give our inputs to the model.
2. **Hidden layer:** Hidden layer performs all sort of computation on the features entered through the input layer and transfer the result to the output layer.
3. **Output layer:** This layer gives the output which is being performed by the network to the outer world.

Back propagation: Back propagation plays an important role in Machine Learning. It is the central mechanism by which neural networks learn. It works like messenger telling the network whether or not the net made any mistake when it made a prediction. In other words it is a mathematical tool for improving the accuracy of model. It works until and unless the desired output matched the actual output.

Error rate= $\sum\ 1/2\ (\text{actual output} - \text{desired output}\ )\ \ \hat{}2$

Correct error= we need to change the weight i.e., (input 1 * w2 + input2 * w1) +w3

New weight= old weight – learning rate x (derivative of error with respect to weight)

W3 = W3 – learning rate (derivative of error / derivative of W3)

Update of W3 = W3 – learning rate x W3 - δh1

A small explanation of neural network techniques:

Dropout: it is a technique to omit the over fitting, during each iteration of GD we draw a randomly selected nodes. Drop the node which does not exist.

Dense: Dense layer is the regular deeply connected neural network layer; it means all the neurons in a layer are connected to those in the next layer. It is the most common and frequently used layer.

Learning rate: In machine learning and statistics, the learning rate is a tuning parameter in an optimization algorithm that determines the step size at each iteration while moving towards a minimum loss function.

OPTIMIZATION Algorithm: The process of minimizing any mathematical expression is called optimization. Optimizers are the algorithms used to change the attributes of the neural networks such as weights and learning rate to reduce the losses. It is used to solve optimization problems by minimizing the function. It can also update the various parameters that can reduce the loss in much less effort. Suppose in a neural network W is a vector called weights and B is a scalar called bias. The weights and bias are called the parameters of the model. All we need to do is estimate the value of W and B from the given set of data such that the resultant hypothesis produces the least cost J which is defined by the following cost.

$$\boldsymbol{J(W,B) = 1/2\mathrm{m}\sum_{i=1}^{m}(\mathrm{Yi} + \mathrm{h}(\mathrm{Xi}))\hat{}2}$$

Where m is the number of data points in the given dataset. This cost function is also called **Mean Squared Error.** For finding the optimized value of the parameters for which J is minimum, optimization algorithm is being used.

ADAM: Adam is a replacement optimization algorithm for stochastic gradient descent for training deep learning model. **Adaptive Gradient Descent:** It maintains a per-parameter learning rate that improves performance on problems with sparse gradients. **Root Mean Squared Propagation:** It also maintains the per parameter learning rates that are adapted based on the average of recent magnitudes of the gradients for the weight that means how quickly it is changed.

ACTIVITYDIAGRAM

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.

Purpose of Activity Diagrams

The basic purposes of activity diagrams are similar to other four diagrams. It captures the dynamic behaviour of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

Where to Use Activity Diagrams?

The basic usage of activity diagram is similar to other four UML diagrams. The specific usage is to model the control flow from one activity to another. This control flow does not include messages.

Activity diagram is suitable for modeling the activity flow of the system. An application can have multiple systems. Activity diagram also captures these systems and describes the flow from one system to another. This specific usage is not available in other diagrams. These systems can be database, external queues, or any other system.

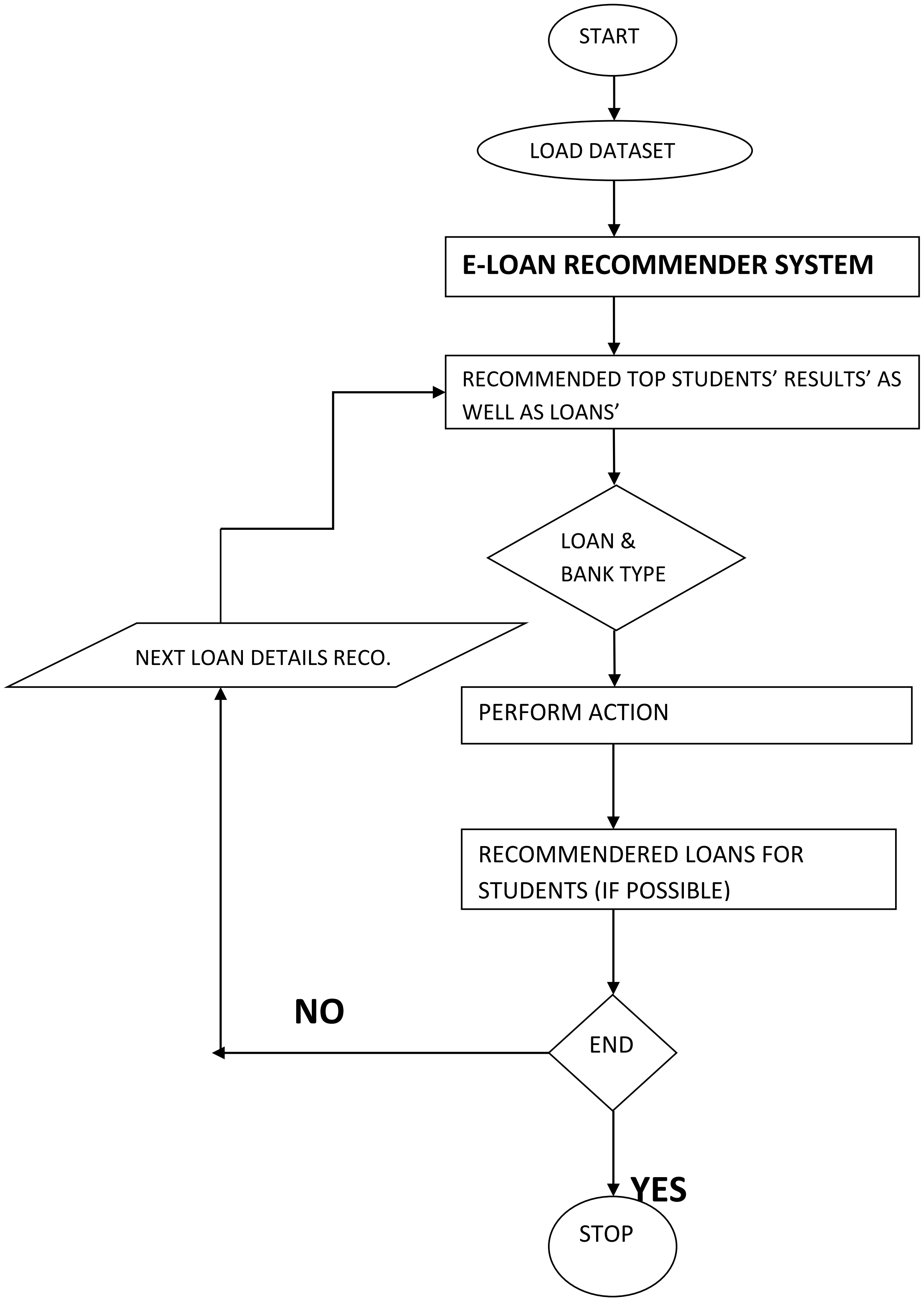

START
LOAD DATASET
E-LOAN RECOMMENDER SYSTEM
RECOMMENDED TOP STUDENTS' RESULTS' AS WELL AS LOANS'
LOAN & BANK TYPE
NEXT LOAN DETAILS RECO.
PERFORM ACTION
RECOMMENDERED LOANS FOR STUDENTS (IF POSSIBLE)
NO
END
YES
STOP

Fig: Activity Diagram