ISSN:0975-9646

Machine Learning Algorithms: A Review

Ayon Dey

Department of CSE, Gautam Buddha University,
Greater Noida, Uttar Pradesh, India

***Abstract* – In this paper, various machine learning algorithms have been discussed. These algorithms are used for various purposes like data mining, image processing, predictive analytics, etc. to name a few. The main advantage of using machine learning is that, once an algorithm learns what to do with data, it can do its work automatically.**

***Keywords* – Machine learning, algorithms, pseudo code**

I. INTRODUCTION

Machine learning is used to teach machines how to handle the data more efficiently. Sometimes after viewing the data, we cannot interpret the pattern or extract information from the data. In that case, we apply machine learning [1]. With the abundance of datasets available, the demand for machine learning is in rise. Many industries from medicine to military apply machine learning to extract relevant information.

The purpose of machine learning is to learn from the data. Many studies have been done on how to make machines learn by themselves [2] [3]. Many mathematicians and programmers apply several approaches to find the solution of this problem. Some of them are demonstrated in Fig. 1.

All the techniques of machine learning are explained in Section 2. Section 3 concludes this paper.

II. TYPES OF LEARNING

A. *Supervised Learning*

The supervised machine learning algorithms are those algorithms which needs external assistance. The input dataset is divided into train and test dataset. The train dataset has output variable which needs to be predicted or classified. All algorithms learn some kind of patterns from the training dataset and apply them to the test dataset for prediction or classification [4]. The workflow of supervised machine learning algorithms is given in Fig. 2. Three most famous supervised machine learning algorithms have been discussed here.

1) Decision Tree: Decision trees are those type of trees which groups attributes by sorting them based on their values. Decision tree is used mainly for classification purpose. Each tree consists of nodes and branches. Each nodes represents attributes in a group that is to be classified and each branch represents a value that the node can take [4]. An example of decision tree is given in Fig. 3.

The pseudo code for Decision tree is described in Fig. 4; where *S*, *A* and *y* are training set, input attribute and target attribute respectively.

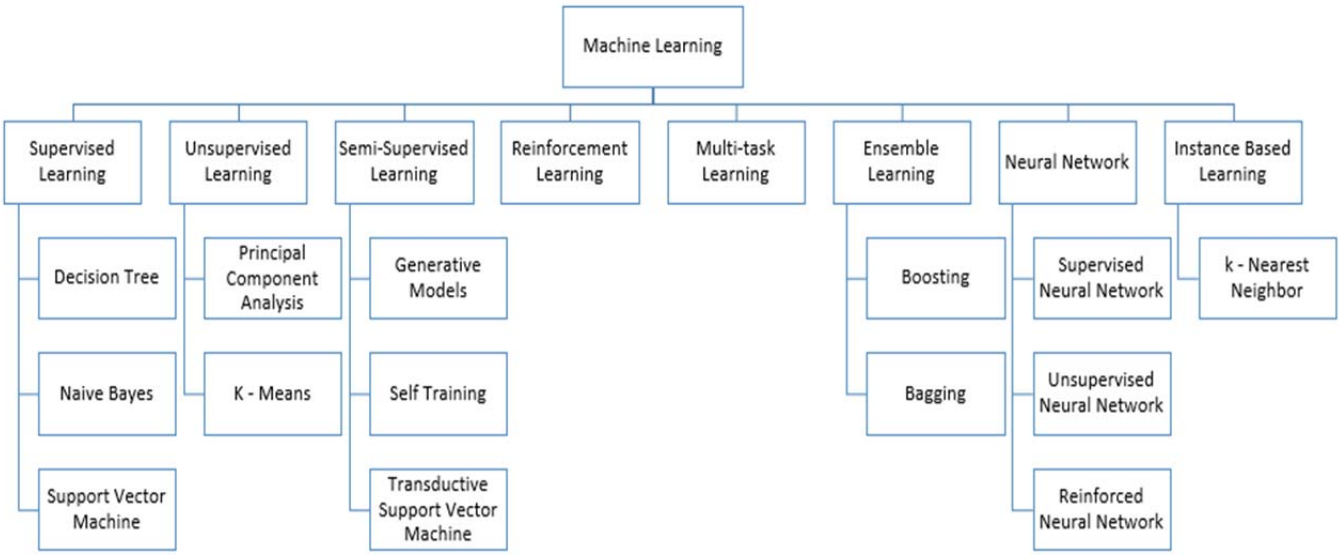

Fig. 1. Types of Learning [2] [3]

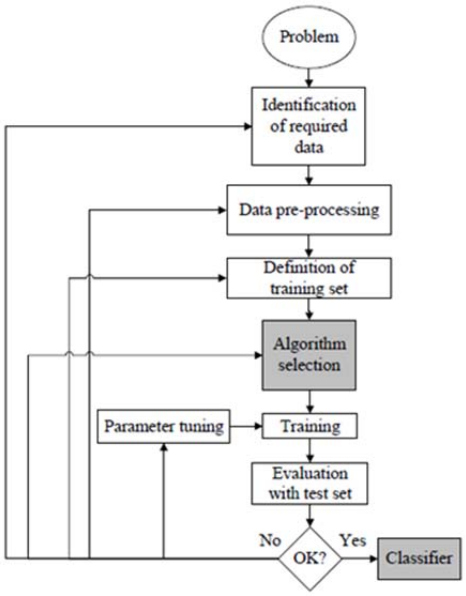

Fig. 2. Workflow of supervised machine learning algorithm [4]

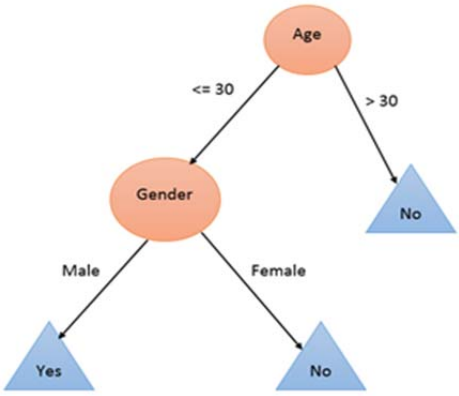

Fig. 3. Decision Tree [5]

2) Naïve Bayes: Naïve Bayes mainly targets the text classification industry. It is mainly used for clustering and classification purpose [6]. The underlying architecture of Naïve Bayes depends on the conditional probability. It creates trees based on their probability of happening. These trees are also known as Bayesian Network. An example of the network is given in Fig. 5. The pseudo code is given in Fig. 6.

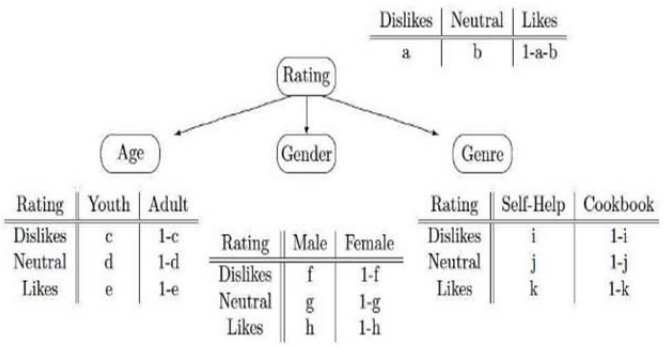

Fig. 5. An Example of Bayesian Network [7]

procedure $DTInducer(S, A, y)$
1: $T = TreeGrowing(S, A, y)$
2: Return TreePruning(S,T)

procedure $TreeGrowing(S, A, y)$
1: Create a tree T
2: **if** One of the Stopping Criteria is fulfilled **then**
3: Mark the root node in T as a leaf with the most common value of y in S as the class.
4: **else**
5: Find a discrete function $f(A)$ of the input attributes values such that splitting S according to $f(A)$'s outcomes $(v_1,\ldots,v_n)$ gains the best splitting metric.
6: **if** best splitting metric $\geq$ treshold **then**
7: Label the root node in T as $f(A)$
8: **for** each outcome v_i of f(A) **do**
9: $Subtree_i = TreeGrowing(\sigma_{f(A)=v_i}S, A, y)$.
10: Connect the root node of T to $Subtree_i$ with an edge that is labelled as v_i
11: **end for**
12: **else**
13: Mark the root node in T as a leaf with the most common value of y in S as the class.
14: **end if**
15: **end if**
16: Return T

procedure $TreePruning(S, T, y)$
1: **repeat**
2: Select a node t in T such that pruning it maximally improve some evaluation criteria
3: **if** $t \neq \emptyset$ **then**
4: $T = pruned(T, t)$
5: **end if**
6: **until** t=Ø
7: Return T

Fig. 4. Pseudo code for Decision Tree [5]

INPUT: training set T, hold-out set H, initial number of components k_0, and convergence thresholds δ_{EM} and δ_{Add}.

Initialize M with one component.
$k \leftarrow k_0$
repeat
Add k new mixture components to M, initialized using k random examples from T.
Remove the k initialization examples from T.
repeat
E-step: Fractionally assign examples in T to mixture components, using M.
M-step: Compute maximum likelihood parameters for M, using the filled-in data.
If $\log P(H|M)$ is best so far, save M in M_{best}.
Every 5 cycles, prune low-weight components of M.
until $\log P(H|M)$ fails to improve by ratio δ_{EM}.
$M \leftarrow M_{best}$
Prune low weight components of M.
$k \leftarrow 2k$
until $\log P(H|M)$ fails to improve by ratio δ_{Add}.
Execute E-step and M-step twice more on M_{best}, using examples from both H and T.
Return M_{best}.

Fig. 6. Pseudo code for Naïve Bayes [6]

3) Support Vector Machine: Another most widely used state-of-the-art machine learning technique is Support Vector Machine (SVM). It is mainly used for classification. SVM works on the principle of margin calculation. It basically, draw margins between the classes. The margins are drawn in such a fashion that the distance between the

margin and the classes is maximum and hence, minimizing the classification error. An example of working and pseudo code of SVM is given in Fig. 7 and Fig. 8, respectively.

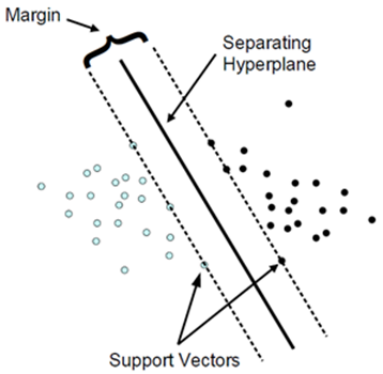

Fig. 7. Working of Support Vector Machine [8]

INPUT: S, λ, T, k
INITIALIZE: Choose $\mathbf{w}_1$ s.t. $\|\mathbf{w}_1\| \leq 1/\sqrt{\lambda}$
FOR $t = 1, 2, \ldots, T$
Choose $A_t \subseteq S$, where $|A_t| = k$
Set $A_t^+ = \{(\mathbf{x}, y) \in A_t : y \langle \mathbf{w}_t, \mathbf{x} \rangle < 1\}$
Set $\eta_t = \frac{1}{\lambda t}$
Set $\mathbf{w}_{t+\frac{1}{2}} = (1 - \eta_t \lambda)\mathbf{w}_t + \frac{\eta_t}{k} \sum_{(\mathbf{x},y) \in A_t^+} y\,\mathbf{x}$
Set $\mathbf{w}_{t+1} = \min\left\{1, \frac{1/\sqrt{\lambda}}{\|\mathbf{w}_{t+\frac{1}{2}}\|}\right\} \mathbf{w}_{t+\frac{1}{2}}$
OUTPUT: $\mathbf{w}_{T+1}$

Fig. 8. Pseudo code for Support Vector machine [9]

B. *Unsupervised Learning*

The unsupervised learning algorithms learns few features from the data. When new data is introduced, it uses the previously learned features to recognize the class of the data. It is mainly used for clustering and feature reduction. An example of workflow of unsupervised learning is given in Fig. 9.

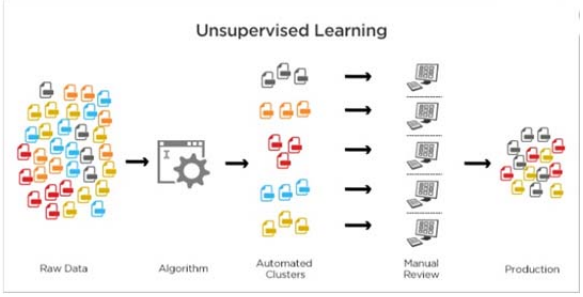

Fig. 9. Example of Unsupervised Learning [10]

The two main algorithms for clustering and dimensionality reduction techniques are discussed below.

1) K-Means Clustering: Clustering or grouping is a type of unsupervised learning technique that when initiates, creates groups automatically. The items which possesses similar characteristics are put in the same cluster. This algorithm is called k-means because it creates k distinct clusters. The mean of the values in a particular cluster is the center of that cluster [9]. A clustered data is represented in Fig. 10. The algorithm for k-means is given in Fig. 11.

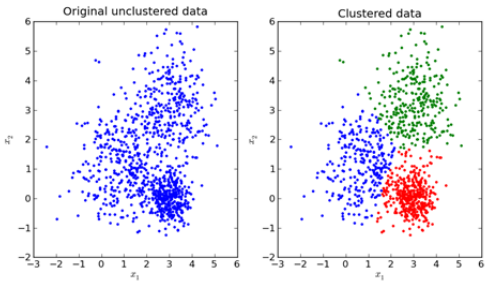

Fig. 10. K-Means Clustering [12]

function Direct-k-means()

Initialize k prototypes $(w_1, \ldots, w_k)$ such that $w_j = i_l,\ j \in \{1, \ldots, k\},\ l \in \{1, \ldots, n\}$

Each cluster C_j is associated with prototype w_j

Repeat

for each input vector i_l, where $l \in \{1, \ldots, n\}$, *do*

Assign i_l to the cluster C_{j*} with nearest prototype w_{j*} (i.e., $|\, i_l - w_{j*} \,| \leq |\, i_l - w_j \,|,\ j \in \{1, \ldots, k\}$)

for each cluster C_j, where $j \in \{1, \ldots, k\}$, *do*

Update the prototype w_j to be the centroid of all samples currently in C_j, so that $w_j = \sum_{i_l \in C_j} i_l / |\, C_j \,|$

Compute the error function:

$$E = \sum_{j=1}^{k} \sum_{i_l \in C_j} |\, i_l - w_j \,|^2$$

Until E does not change significantly or cluster membership no longer changes

Fig. 11. Pseudo code for k-means clustering [13]

2) Principal Component Analysis

In Principal Component Analysis or PCA, the dimension of the data is reduced to make the computations faster and easier. To understand how PCA works, let's take an example of 2D data. When the data is being plot in a graph, it will take up two axes. PCA is applied on the data, the data then will be 1D. This is explained in Fig. 12. The pseudo code for PCA is discussed in Fig. 13.

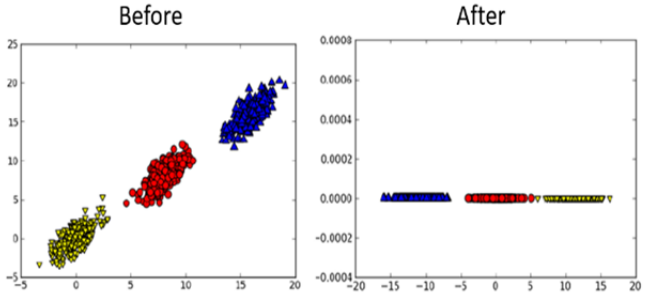

Fig. 12. Visualization of data before and after applying PCA [11]

$$
\begin{aligned}
&\mathbf{R} \leftarrow \mathbf{X} \\
&\text{for}(k = 0, ..., K-1) \text{ do} \\
&\quad \{ \\
&\quad \lambda = 0 \\
&\quad \mathbf{T}^{(k)} \leftarrow \mathbf{R}^{(k)} \\
&\quad \text{for}(j = 0, ..., J) \text{ do} \\
&\quad\quad \{ \\
&\quad\quad \mathbf{P}^{(k)} \leftarrow \mathbf{R}^T \mathbf{T}^{(k)} \\
&\quad\quad \mathbf{P}^{(k)} \leftarrow \mathbf{P}^{(k)} \left\| \mathbf{P}^{(k)} \right\|^{-1} \\
&\quad\quad \mathbf{T}^{(k)} \leftarrow \mathbf{R}\mathbf{P}^{(k)} \\
&\quad\quad \lambda' \leftarrow \left\| \mathbf{T}^{(k)} \right\| \\
&\quad\quad \text{if}(|\lambda' - \lambda| \leq \varepsilon) \text{ then break} \\
&\quad\quad \lambda \leftarrow \lambda' \\
&\quad\quad \} \\
&\quad \mathbf{R} \leftarrow \mathbf{R} - \mathbf{T}^{(k)} (\mathbf{P}^{(k)})^T \\
&\quad \} \\
&\text{return } \mathbf{T}, \mathbf{P}, \mathbf{R}
\end{aligned}
$$

Fig. 13. Pseudo code for PCA [14]

C. Semi - Supervised Learning

Semi – supervised learning algorithms is a technique which combines the power of both supervised and unsupervised learning. It can be fruit-full in those areas of machine learning and data mining where the unlabeled data is already present and getting the labeled data is a tedious process [15]. There are many categories of semi-supervised learning [16]. Some of which are discussed below:

1) Generative Models: Generative models are one of the oldest semi-supervised learning method assumes a structure like $p(x,y) = p(y)p(x|y)$ where $p(x|y)$ is a mixed distribution e.g. Gaussian mixture models. Within the unlabeled data, the mixed components can be identifiable. One labeled example per component is enough to confirm the mixture distribution.

2) Self-Training: In self-training, a classifier is trained with a portion of labeled data. The classifier is then fed with unlabeled data. The unlabeled points and the predicted labels are added together in the training set. This procedure is then repeated further. Since the classifier is learning itself, hence the name self-training.

3) Transductive SVM: Transductive support vector machine or TSVM is an extension of SVM. In TSVM, the labeled and unlabeled data both are considered. It is used to label the unlabeled data in such a way that the margin is maximum between the labeled and unlabeled data. Finding an exact solution by TSVM is a NP-hard problem.

D. Reinforcement Learning

Reinforcement learning is a type of learning which makes decisions based on which actions to take such that the outcome is more positive. The learner has no knowledge which actions to take until it's been given a situation. The action which is taken by the learner may affect situations and their actions in the future. Reinforcement learning solely depends on two criteria: trial and error search and delayed outcome [17]. The general model [18] for reinforcement learning is depicted in Fig. 14.

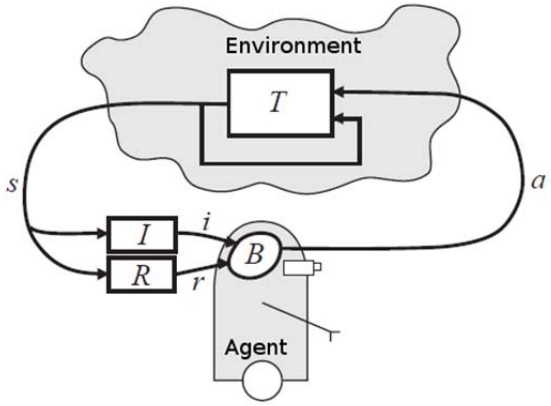

Fig. 14. The Reinforcement Learning Model [18]

In the figure, the agent receives an input *i*, current state *s*, state transition *r* and input function *I* from the environment. Based on these inputs, the agent generates a behavior *B* and takes an action *a* which generates an outcome.

E. Multitask Learning

Multitask learning has a simple goal of helping other learners to perform better. When multitask learning algorithms are applied on a task, it remembers the procedure how it solved the problem or how it reaches to the particular conclusion. The algorithm then uses these steps to find the solution of other similar problem or task. This helping of one algorithm to another can also be termed as inductive transfer mechanism. If the learners share their experience with each other, the learners can learn concurrently rather than individually and can be much faster [19].

F. Ensemble Learning

When various individual learners are combined to form only one learner then that particular type of learning is called ensemble learning. The individual learner may be Naïve Bayes, decision tree, neural network, etc. Ensemble learning is a hot topic since 1990s. It has been observed that, a collection of learners is almost always better at doing a particular job rather than individual learners [20]. Two popular Ensemble learning techniques are given below [21]:

1) Boosting: Boosting is a technique in ensemble learning which is used to decrease bias and variance. Boosting creates a collection of weak learners and convert them to one strong learner. A weak learner is a classifier which is barely correlated with true classification. On the other hand, a strong learner is a type of classifier which is strongly correlated with true classification [21]. The pseudo code for AdaBoost (which is most popular example of boosting) is give in Fig. 15.

Input: Data set $\mathcal{D} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_m, y_m)\}$;
Base learning algorithm $\mathcal{L}$;
Number of learning rounds T.

Process:
$D_1(i) = 1/m$.
for $t = 1, \cdots, T$:
$h_t = \mathcal{L}(\mathcal{D}, D_t)$;
$\epsilon_t = \Pr_{i \sim D_i}[h_t(\boldsymbol{x}_i \neq y_i)]$;
$\alpha_t = \frac{1}{2}\ln \frac{1-\epsilon_t}{\epsilon_t}$;

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} \exp(-\alpha_t) & \text{if } h_t(\boldsymbol{x}_i) = y_i \\ \exp(\alpha_t) & \text{if } h_t(\boldsymbol{x}_i) \neq y_i \end{cases} = \frac{D_t(i)\exp(-\alpha_t y_i h_t(\boldsymbol{x}_i))}{Z_t}$$

end.

Output: $H(\boldsymbol{x}) = \text{sign}\,(f(\boldsymbol{x})) = \text{sign} \sum_{t=1}^{T} \alpha_t h_t(\boldsymbol{x})$

Fig. 15. Pseudo code for AdaBoost [21]

2) Bagging: Bagging or bootstrap aggregating is applied where the accuracy and stability of a machine learning algorithm needs to be increased. It is applicable in classification and regression. Bagging also decreases variance and helps in handling overfitting [23]. The pseudo code for bagging in given in Fig. 16.

Input: Data set $\mathcal{D} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_m, y_m)\}$;
Base learning algorithm $\mathcal{L}$;
Number of learning rounds T.

Process:
for $t = 1, \cdots, T$:
$\mathcal{D}_t = Bootstrap(\mathcal{D})$;
$h_t = \mathcal{L}(\mathcal{D}_t)$
end.

Output: $H(\boldsymbol{x}) = \text{argmax}_{y \in \mathcal{Y}} \sum_{t=1}^{T} 1(y = h_t(\boldsymbol{x}))$

Fig. 16. Pseudo code for Bagging [21]

G. Neural Network Learning

The neural network (or artificial neural network or ANN) is derived from the biological concept of neurons. A neuron is a cell like structure in a brain. To understand neural network, one must understand how a neuron works. A neuron has mainly four parts (see Fig. 17). They are dendrites, nucleus, soma and axon.

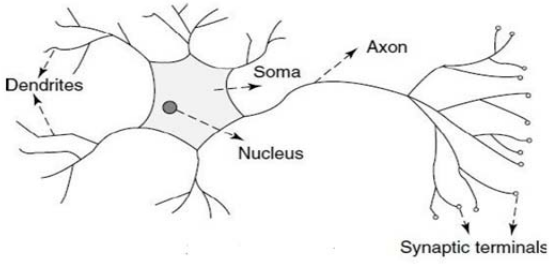

Fig. 17. A Neuron [24]

The dendrites receive electrical signals. Soma processes the electrical signal. The output of the process is carried by the axon to the dendrite terminals where the output is sent to next neuron. The nucleus is the heart of the neuron. The inter-connection of neuron is called neural network where electrical impulses travel around the brain.

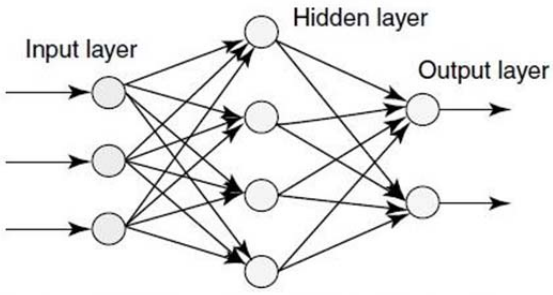

Fig. 18. Structure of an Artificial Neural Network [24]

An artificial neural network behaves the same way. It works on three layers. The input layer takes input (much like dendrites). The hidden layer processes the input (like soma and axon). Finally, the output layer sends the calculated output (like dendrite terminals) [24]. There are basically three types of artificial neural network: supervised, unsupervised and reinforcement [25].

1) Supervised Neural Network: In the supervised neural network, the output of the input is already known. The predicted output of the neural network is compared with the actual output. Based on the error, the parameters are changed, and then fed into the neural network again. Fig. 19 will summarize the process. Supervised neural network is used in feed forward neural network.

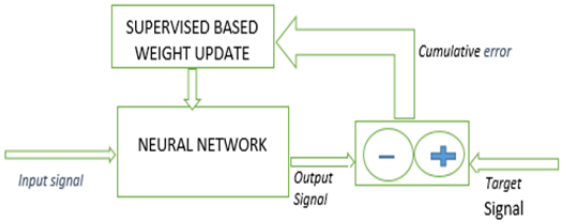

Fig. 19. Supervised Neural Network [25]

2) Unsupervised Neural Network: Here, the neural network has no prior clue about the output the input. The main job of the network is to categorize the data according to some similarities. The neural network checks the correlation between various inputs and groups them. The schematic diagram is shown in Fig. 20.

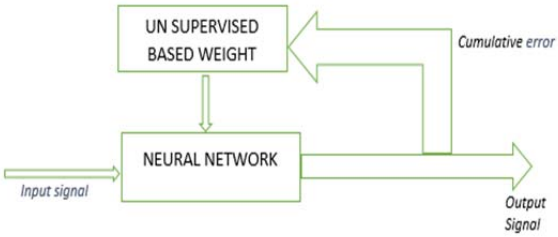

Fig. 20. Unsupervised Neural Network [25]

3) Reinforced Neural Network: In reinforced neural network, the network behaves as if a human communicates with the environment. From the environment, a feedback has been provided to the network acknowledging the fact that whether the decision taken by the network is right or

wrong. If the decision is right, the connections which points to that particular output is strengthened. The connections are weakened otherwise. The network has no previous information about the output. Reinforced neural network is represented in Fig. 21.

4)

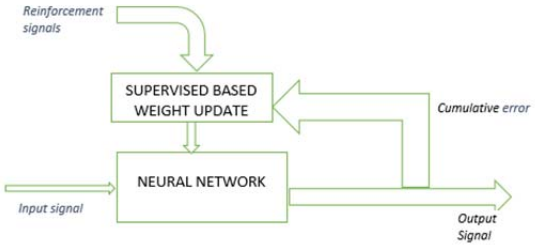

Fig. 21. Reinforced Neural Network [25]

H. Instance-Based Learning

In instance-based learning, the learner learns a particular type of pattern. It tries to apply the same pattern to the newly fed data. Hence the name instance-based. It is a type of lazy learner which waits for the test data to arrive and then act on it together with training data. The complexity of the learning algorithm increases with the size of the data. Given below is a well-known example of instance-based learning which is k-nearest neighbor [26].

1) K-Nearest Neighbor: In k-nearest neighbor (or KNN), the training data (which is well-labeled) is fed into the learner. When the test data is introduced to the learner, it compares both the data. *k* most correlated data is taken from training set. The majority of *k* is taken which serves as the new class for the test data [27]. The pseudo code for KNN is given in Fig. 22.

Let $W = \{x_1, x_2, \cdots, x_n\}$ be a set of n labeled samples. The algorithm is as follows:

```
BEGIN
  Input y, of unknown classification.
  Set K, 1 ⩽ K ⩽ n.
  Initialize i = 1.
  DO UNTIL (K-nearest neighbors found)
    Compute distance from y to x_i.
    IF (i ⩽ K) THEN
      Include x_i in the set of K-nearest neighbors
    ELSE IF (x_i is closer to y than any previous nearest
      neighbor) THEN
      Delete farthest in the set of K-nearest neighbors
      Include x_i in the set of K-nearest neighbors.
    END IF
    Increment i.

  END DO UNTIL
  Determine the majority class represented in the set of K-
  nearest neighbors.
  IF (a tie exists) THEN
    Compute sum of distances of neighbors in each class
    which tied.
    IF (no tie occurs) THEN
      Classify y in the class of minimum sum
    ELSE
      Classify y in the class of last minimum found.
    END IF
  ELSE
    Classify y in the majority class.
  END IF
END
```

Fig. 22. Pseudo code for K-Nearest Neighbor [28]

III. CONCLUSION

This paper surveys various machine learning algorithms. Today each and every person is using machine learning knowingly or unknowingly. From getting a recommended product in online shopping to updating photos in social networking sites. This paper gives an introduction to most of the popular machine learning algorithms.

REFERENCES

[1] W. Richert, L. P. Coelho, "*Building Machine Learning Systems with Python*", Packt Publishing Ltd., ISBN 978-1-78216-140-0
[2] M. Welling, "*A First Encounter with Machine Learning*"
[3] M. Bowles, "*Machine Learning in Python: Essential Techniques for Predictive Analytics*", John Wiley & Sons Inc., ISBN: 978-1-118-96174-2
[4] S.B. Kotsiantis, "*Supervised Machine Learning: A Review of Classification Techniques*", Informatica 31 (2007) 249-268
[5] L. Rokach, O. Maimon, "*Top – Down Induction of Decision Trees Classifiers – A Survey*", IEEE Transactions on Systems,
[6] D. Lowd, P. Domingos, "Naïve Bayes Models for Probability Estimation"
[7] https://webdocs.cs.ualberta.ca/~greiner/C-651/Homework2_Fall2008.html
[8] D. Meyer, "Support Vector Machines – The Interface to libsvm in package e1071", August 2015
[9] S. S. Shwartz, Y. Singer, N. Srebro, "*Pegasos: Primal Estimated sub - Gradient Solver for SVM*", Proceedings of the 24th International Conference on Machine Learning, Corvallis, OR, 2007
[10] http://www.simplilearn.com/what-is-machine-learning-and-why-it-matters-article
[11] P. Harrington, "*Machine Learning in action*", Manning Publications Co., Shelter Island, New York, 2012
[12] http://pypr.sourceforge.net/kmeans.html
[13] K. Alsabati, S. Ranaka, V. Singh, "*An efficient k-means clustering algorithm*", Electrical Engineering and Computer Science, 1997
[14] M. Andrecut, "*Parallel GPU Implementation of Iterative PCA Algorithms*", Institute of Biocomplexity and Informatics, University of Calgary, Canada, 2008
[15] X. Zhu, A. B. Goldberg, "*Introduction to Semi – Supervised Learning*", Synthesis Lectures on Artificial Intelligence and Machine Learning, 2009, Vol. 3, No. 1, Pages 1-130
[16] X. Zhu, "*Semi-Supervised Learning Literature Survey*", Computer Sciences, University of Wisconsin-Madison, No. 1530, 2005
[17] R. S. Sutton, "*Introduction: The Challenge of Reinforcement Learning*", Machine Learning, 8, Page 225-227, Kluwer Academic Publishers, Boston, 1992
[18] L. P. Kaelbing, M. L. Littman, A. W. Moore, "*Reinforcement Learning: A Survey*", Journal of Artificial Intelligence Research, 4, Page 237-285, 1996
[19] R. Caruana, "*Multitask Learning*", Machine Learning, 28, 41-75, Kluwer Academic Publishers, 1997
[20] D. Opitz, R. Maclin, "*Popular Ensemble Methods: An Empirical Study*", Journal of Artificial Intelligence Research, 11, Pages 169-198, 1999
[21] Z. H. Zhou, "*Ensemble Learning*", National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
[22] https://en.wikipedia.org/wiki/Boosting_(machine_learning)
[23] https://en.wikipedia.org/wiki/Bootstrap_aggregating
[24] V. Sharma, S. Rai, A. Dev, "*A Comprehensive Study of Artificial Neural Networks*", International Journal of Advanced Research in Computer Science and Software Engineering, ISSN 2277128X, Volume 2, Issue 10, October 2012
[25] S. B. Hiregoudar, K. Manjunath, K. S. Patil, "*A Survey: Research Summary on Neural Networks*", International Journal of Research in Engineering and Technology, ISSN: 2319 1163, Volume 03, Special Issue 03, pages 385-389, May, 2014
[26] https://en.wikipedia.org/wiki/Instance-based_learning
[27] P. Harrington, "Machine Learning in Action", Manning Publications Co., Shelter Island, New York, ISBN 9781617290183, 2012
[28] J. M. Keller, M. R. Gray, J. A. Givens Jr., "*A Fuzzy K-Nearest Neighbor Algorithm*", IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-15, No. 4, August 1985