Algorithms and Applications of Supervised Learning

Kingshuk Mukherjee, 544
St. Xavier's College (Autonomous), Kolkata

***Abstract*—Supervised learning is by far the most common area of machine learning. In supervised learning, the goal is to learn how training inputs and training targets relate to each other. A number of different algorithms have been developed in this field of supervised learning. This paper discusses the basics of some of the common algorithms used in the field and their applications.**

***Keywords*—machine learning, supervised, algorithms**

I. INTRODUCTION

Machine learning is the most popular and successful subfield of Artificial Intelligence [1]. It is a mode of approaching a problem of automation (playing a game, classifying pictures etc.) where the system being developed is fed the 'data' and the correct 'answers' to the data as input, and is expected to figure out the method of finding the 'rules' for giving correct answers to the data by itself. This is what separates machine learning from classical artificial intelligence programming, in which the 'rules' are hard-coded into the system at the outset. Thus, the main purpose of machine learning is to 'learn' to predict correct outcomes from the data in this way. A lot of studies have been conducted in order to enable this [2][3].

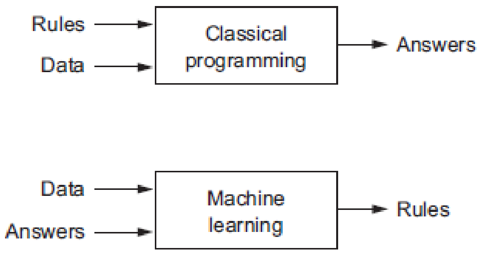

Fig. 1. Machine Learning Paradigm [1].

The most important subfields of machine learning are as follows:

- Supervised learning—By far the most widespread application of machine learning. It consists of learning to map input data to a set of known targets, given a set of examples which are already correctly annotated with their targets by humans [1].
- Unsupervised learning—Consists of trying to draw inferences and find hidden patterns from input data without the help of any targets or labels [6]. This approach is extensively used in data visualizations and data analytics.
- Reinforcement learning—In reinforcement learning, an agent receives information about its environment and learns to choose actions that will maximize some reward. For instance, a neural network that "looks" at a videogame screen and outputs game actions in order to maximize its score can be trained via reinforcement learning.[1]

Supervised learning can be applied when we cannot interpret the pattern or extract information from the data [4]. It requires three things:

- Input data points—If the task is text recognition, data points could be text files; if the task is image classification, they could be pictures.
- Examples of expected output—In an image classification scheme, the outputs might be text labels such as 'dog', 'cat' etc.
- A way to measure whether the algorithm is doing a good job—This is necessary in order to determine the distance between the algorithm's current output and its expected output. The measurement is used as a feedback signal to adjust the way the algorithm works.

The idea is for the training set to "learn" from a set of labeled examples in the training set so that it can identify unlabeled examples in the test set with the highest possible accuracy. That is, the goal of the learner is to develop a rule, a program, or a procedure that classifies new examples (in the test set) by analyzing examples it has been given that already have a class label. For example, a training set might consist of images of different types of fruit (say, peaches and nectarines), where the identity of the fruit in each image is given to the learner. The test set would then consist of more unidentified pieces of fruit, but from the same classes. The goal is for the learner to develop a rule that can identify the elements in the test set. There are many different approaches that attempt to build the best possible method of classifying examples of the test set by using the data given in the training set.[13]

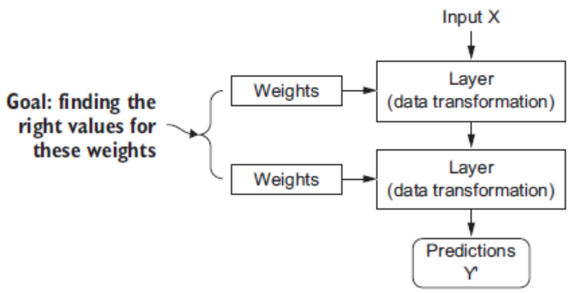

Fig. 2. Basic supervised learning flowchart in a neural network [1].

The model is exposed to known examples of inputs and outputs that are annotated correctly, for example a set of pictures which are correctly classified as being those of cats and dogs. The pictures with their labels are fed to the model and it gets increasingly better at classifying unknown cat/dog pictures correctly by itself. The problems can be solved by building a model that is a good representation of a selected dataset [5]. A representation of a data point is a way of looking at that data point that will help solve the problem at hand. For example, an image can be represented in the RGB (red-green-blue) format or the HSV (hue-saturation-value) format depending on the way the images need to be classified.

An overview of the most typical use case of supervised learning (a neural network), its applications, and two most common supervised learning algorithms are provided in section II. Section III concludes the paper.

II. Supervised Learning Algorithms

A typical supervised learning model in a neural network has one or more layers stacked on top of each other. A layer is a construct through which the input data passes, and the output is a more useful representation of the data which will help us map the data to the labels correctly. The specification of what a layer does to its input data is defined by its weights, which are actually a set of numbers. In this context, 'learning' means finding a set of values for the weights of all layers in a network, such that the network will correctly map example inputs to their associated targets.

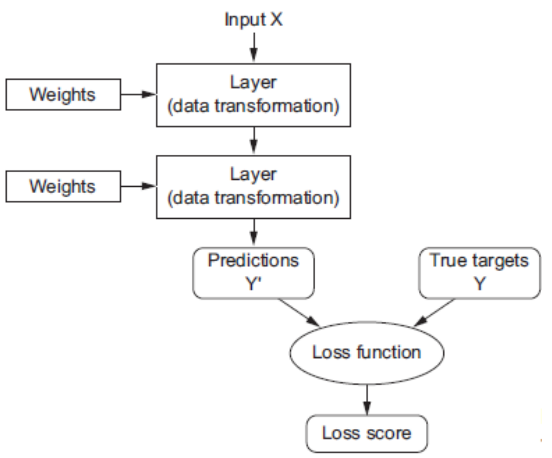

Fig. 3. More Detailed Flowchart Showing Loss Function [1].

To measure how far the output of the model is than expected, a loss function is used. It takes the predictions of the network and the true target to compute a distance score, thus providing a measure of how well the network worked.

This score is then used as a feedback signal to adjust the value of the weights a little in the direction of the more correct output.

Initially, the model starts out with random values of weights, so the outputs are not accurate at all, and the loss score is very high. But with each example the model processes, the weights are adjusted a little bit to the correct direction. This is the process of training the network. A model with minimal loss is one for which the outputs match the required targets as closely as possible.

A brief overview of the main application areas of supervised learning is provided below.

- Binary classification—In the field of information extraction and retrieval, binary classification is the process of classifying given document/account on the basis of predefined classes [7]. The number of classes is typically two. This is the widest application area of supervised learning. An example might be an image classifier that classifies images based on whether the image contains a human or not. The labels in this case might be 'human' and 'nonhuman'.
- Multiclass classification—In multiclass classification, the number of classes are more than two. An image classifier that classifies images on three separate labels of 'dog', 'cat', and 'bird' is an example of multiclass classification.
- Scalar Regression—In scalar regression, the main task is to try to predict a single continuous value. Predicting house prices, or age of a person based on facial features, are typical examples of scalar regression tasks.

What follows is a discussion of two of the most common algorithms that employ supervised learning.

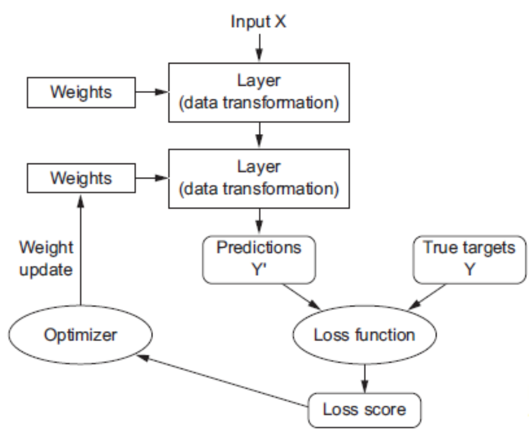

Fig. 4. Complete Flowchart Showing all the important components [1].

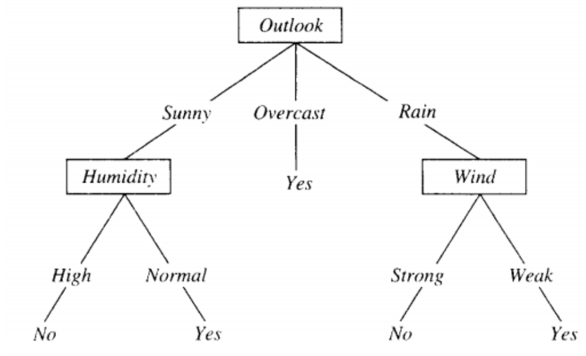

Fig. 5. Typical Decision Tree[11]

A. Decision Trees

Decision trees are those types 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 node represents attributes in a group that is to be classified and each branch represents a value that the node can take [10]. An example of decision tree is given in Fig. 5. In the figure, the tree classifies mornings according to whether or not they are suitable for playing tennis.

An instance is classified by starting at the root node, testing the attribute specified by the node, then moving down the tree branch corresponding to the value of the attribute in the given example. The process is then repeated for the subtree rooted at the new node.

The instance "outlook=sunny, temperature=hot, humidity=high, wind=strong" would be sorted along the leftmost branch of the decision tree of Fig. 5. and would therefore be classified as a negative instance, i.e., "playtennis=no".

The decision tree is trained with a set of attributes with their corresponding values, according to which it decides what attributes to keep at what nodes of the tree. This is the 'learning' process of the decision tree. The basic algorithm consists of 3 steps:

Step 1. Each instance attribute is evaluated using a statistical test to determine how well it 'alone' classifies the training examples. The best attribute in this regard is selected and kept at the root. Several statistical methods exist which can facilitate this, but they are beyond the scope of this review.

Step 2. A descendant of the root node is then created for each possible value of this attribute, and the training examples are sorted to the appropriate descendant node.

Step 3. Entire process is repeated from step 1 using the training examples associated with each descendant node to select the best attribute (using the same statistical method) to test at that point of the tree.

The algorithm is provided in Fig.6.

ID3(*Examples*, *Target_attribute*, *Attributes*)

Examples are the training examples. Target_attribute is the attribute whose value is to be predicted by the tree. Attributes is a list of other attributes that may be tested by the learned decision tree. Returns a decision tree that correctly classifies the given Examples.

- Create a *Root* node for the tree
- If all *Examples* are positive, Return the single-node tree *Root*, with label = +
- If all *Examples* are negative, Return the single-node tree *Root*, with label = −
- If *Attributes* is empty, Return the single-node tree *Root*, with label = most common value of *Target_attribute* in *Examples*
- Otherwise Begin
  - $A \leftarrow$ the attribute from *Attributes* that best* classifies *Examples*
  - The decision attribute for *Root* $\leftarrow A$
  - For each possible value, v_i, of A,
    - Add a new tree branch below *Root*, corresponding to the test $A = v_i$
    - Let $Examples_{v_i}$ be the subset of *Examples* that have value v_i for A
    - If $Examples_{v_i}$ is empty
      - Then below this new branch add a leaf node with label = most common value of *Target_attribute* in *Examples*
      - Else below this new branch add the subtree ID3($Examples_{v_i}$, *Target_attribute*, *Attributes* − {A}))
- End
- Return *Root*

Fig. 6. Algorithm of Decision Tree Building[11]

B. Linear Regression

The goal of the linear regression, as a part of the family of regression algorithms, is to find relationships and dependencies between variables. [5] It represents a modeling relationship between a continuous scalar dependent variable y (also label or target in machine learning terminology) and one or more (a D-dimensional vector) explanatory variables (also independent variables, input variables, features, observed data, observations, attributes, dimensions, data point, etc.) denoted x using a linear function. In regression analysis the goal is to predict a continuous target variable, whereas another area called classification is predicting a label from a finite set. The model for a multiple regression which involves linear combination of input variables takes the form:

$$y = a0 + a1.x \qquad (1)$$

Linear regression [8] belongs to the category of supervised learning algorithms. It means we train the model on a set of labeled data (training data) and then use the model to predict labels on unlabeled data (testing data).

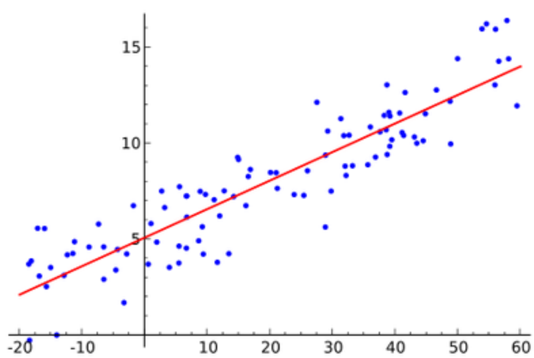

Fig. 6. Visual representation of linear regression [9]

As shown on Fig. 6, the model (red line) is calculated using training data (blue points) where each point has a known label (y axis) to fit the points as accurately as possible by minimizing the value of a chosen loss function. We can then use the model to to predict unknown labels (we only know x value and want to predict y value).

In the case of linear regression, the loss function can be chosen as the Mean Squared Error (MSE) function J, which squares the error difference of each data point, sums all those values, and divides that sum by the number of data points n.

$$minimize \frac{1}{n} \sum_{i=1}^{n} (pred_i - y_i)^2$$

$$J = \frac{1}{n} \sum_{i=1}^{n} (pred_i - y_i)^2$$

Fig. 7. Loss Function J [12].

To update the values of a0 and a1, we find the partial derivatives of the above equation with respect to a0 and a1, multiply those two values by learning rate α, and subtract those values from the initial values of a0 and a1, thus giving us the upgraded values of a0 and a1.

$$a_0 = a_0 - \alpha \cdot \frac{2}{n} \sum_{i=1}^{n} (pred_i - y_i)$$

$$a_1 = a_1 - \alpha \cdot \frac{2}{n} \sum_{i=1}^{n} (pred_i - y_i) \cdot x_i$$

Fig. 8. Getting new values of a0 and a1[12].

As each training session is processed, the MSE is calculated anew and the value of a0 and a1 is updated a little bit to the correct direction by using the above method. This constitutes the training of the model.

```
Step 1. initialize a0=a1=0.0, alpha=0.0001, number_of_epochs=0
Step 2. while (number_of_epochs < 1000)
            y = a0 + a1*x_train
            error = y - y_train
            mean_squre_error = 1/n * sum(error^2)
            a0 = a0 - alpha * 2 * sum(error)/n
            a_1 = a_1 - alpha * 2 * np.sum(error * x_train)/n
Step 3. end
```

Fig. 9. Training Algorithm of Linear Regression.

III. Conclusion

Supervised Learning is one of the more successful approaches of machine learning [5] because the availability of human-labeled data beforehand provides us more robust criteria for optimization.

Machine learning is a rapidly growing research area. Its relevance has increased with the need to automate tasks such as image and text classification, prediction and a host of other applications.

In this paper, a brief overview of machine learning was provided with supervised learning in focus. The two main tasks in which supervised learning is employed are classification and regression. The paper discussed two of the common supervised learning algorithms used in the above two areas, namely Decision Trees and Linear Regression.

An area of concern remains that efficient machine learning systems require tremendous processing power. The more technology develops, the better the algorithms discussed here will function. Because of the rapid progression, there is plenty of space for the developers to work or to improve the supervised learning methods and their algorithms.

References

[1] F. Chollet, “Deep Learning With Python”. Manning Publications, 2017.

[2] M. Welling, “A First Encounter With Machine learning”. Donald Bren School of Information and Computer Science,University of California Irvine, November 4, 2011.

[3] M. Bowles, “Machine Learning in Python: Essential Techniques for Predictive Analytics”, John Wiley & Sons Inc., ISBN: 978-1-118-96174-2.

[4] W. Richert, L. P. Coelho, “Building Machine Learning Systems with Python”, Packt Publishing Ltd., ISBN 978-1-78216-140-0.

[5] Nasteski, Vladimir. (2017). An overview of the supervised machine learning methods. HORIZONS.B. 4. 51-62. 10.20544/HORIZONS.B.04.1.17.P05.

[6] Khanam, Memoona & Mahboob, Tahira & Imtiaz, Warda & Ghafoor, Humaraia & Sehar, Rabeea. (2015). A Survey on Unsupervised Machine Learning Algorithms for Automation, Classification and Maintenance. International Journal of Computer Applications. 119. 34-39. 10.5120/21131-4058.

[7] Kumari, Roshan & Srivastava, Saurabh. (2017). Machine Learning: A Review on Binary Classification. International Journal of Computer Applications. 160. 11-15. 10.5120/ijca2017913083.

[8] C.-J. Lin, R. C. Weng, S. S. Keerthi. (2008) Trust region Newton method for large-scale logistic regression. In Journal of Machine Learning Research, vol. 9.

[9] http://gerardnico.com/wiki/data_mining/simple_regression

[10] S.B. Kotsiantis, “Supervised Machine Learning: A Review of Classification Techniques”, Informatica 31 (2007) 249-268

[11] https://www.cs.princeton.edu/courses/archive/spring07/cos424/papers/mitchell-dectrees.pdf

[12] https://towardsdatascience.com/introduction-to-machine-learning-algorithms-linear-regression-14c4e325882a

[13] Learned-Miller, Erik G. "Introduction to Supervised Learning". Department of Computer Science, University of Massachusetts, Amherst, Amherst, MA 01003. February 17, 2014.