Literature Review Comparison b/t KNN and LSTM

Name : Junaid Ur Rahman
dept. Computer Science
(SE-A)
Iqra National University
Peshawar, Pakistan
junaidurrahman840@gmail.com
(17896)

Abstract

In this paper majorly all the aspects concerning machine learning algorithms namely-K-Nearest Neighbor (KNN), and Long Short Term Memory (LSTM) network have been discussed. The paper answers all relevant questions that may arise during the study of these algorithms ranging from their origin, to their definition, methodologies of execution, real-time applications attached with sufficient novel evidence, followed by the advantages and major trade-offs; lastly an elaborate comparison of their performances on quantitative and qualitative grounds has been presented. To conclude, the paper also highlights the future scope of ML algorithms and artificial intelligence in the coming times and their roles in automation and holistic development.

Keywords—Machine Learning, K-Nearest Neighbor, Long Short-Term Memory.

I. Introduction

1.1 Machine Learning

Machine Learning, an amalgamation of statistical concepts and scientific knowledge of computers, this term was coined by Arthur Samuel back in 1959, and today, it is considered to be a subset or a subpart of Artificial Intelligence (AI), associated to algorithms that permit processors or computers to automatically process and classify new data based on old data and information

1.1.1 Supervised ML

In this category, the machine is provided with labeled sample data intended for training it, based on which it would later be predicting outputs. Following this process, the machine is tested for correct and exact outputs with some random data.

1.1.2 Unsupervised ML

Unsupervised learning enables the machine to learn without any supervision. In unsupervised learning, an unsegregated and unlabeled data set is provided to the machine, and the algorithm is supposed to perform on the data without any supervision.

2. K-nearest neighbour neighbor algorithm (K-NN)

K-nearest-neighbor (K-NN) being one of the most essential and effective algorithms for data segregation is capable of becoming the primary choice for implementation especially when the given data is quite ambiguous. This algorithm was invented back in 1951 by Evelyn Fix and Joseph Hodges for discriminant examination when it was relatively challenging to decide the probabilistic densities by parametric estimation [1]. Further in the year 1967, a couple of characteristics belonging to this algorithm were calculated, for example where ‘k’ = 1 and ‘n’ tends to infinity then the K-NN classification fallacy or error is limited above by two times the error rate of Bayes [2]. Post establishment of such particular characteristics and properties, research and experimentation followed over long periods to count novel rejection approaches [3], improvements for Bayes error rate , procedures relying solely on distance. methods for soft computing and other approaches. The K-NN algorithm is positioned under the supervised type learning technique and is considered one of the easiest-to-use algorithms in Machine Learning. Although it is suited for classifying as well as regressing both, it is predominantly utilized for classifying objects. It is an extremely handy algorithm, used to assign any missing value and to re-sample the data.

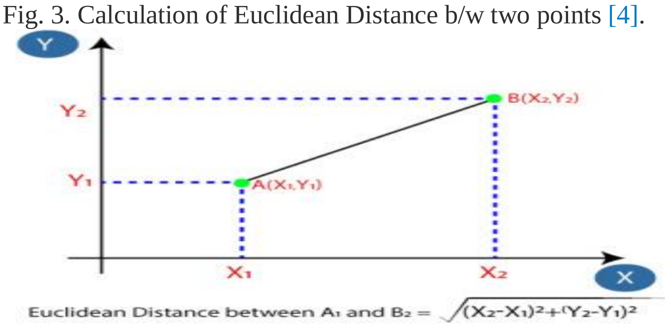
2.1. Working of KNN algorithm

The ‘K’ here in K-NN refers to the count of neighbors of the new data point. Deciding a suitable value for K is the foremost process in this algorithm. For better accuracy, it is imperative that one chooses the accurate value of K, and this process is called parameter tuning. A very low value of K like 1 or 2 can lead to noisy results, whereas, a very high value can create confusion at times, depending on the data set. There is no fixed value for K, however, one of the standard values that K often assumes is ‘5’ i.e., for the majority voting process, the 5 neighbors closest to the new data point are considered. To avoid mistakes and confusion among two classes of data sets, generally, an odd value of K is suitable. Another formula-based calculation for k can be done by this formula:

$$k = \sqrt{n}$$

And, **n** is the overall count of data points.
Followed by that, the Euclidean type distance of the prevailing points in the data set from the new data point is calculated. In order to do so, it is imperative, that the data set be plotted in a graphical manner. Euclidean distance is calculated as shown in Fig. 3.
Upon calculating the values of the Euclidean distances of all the points from the new data point, one should observe the category to which the majority of the nearest neighbors belong (say, at K = 5), and hence after careful computation impute that class to the data point, assigned for classification.
Like in Fig. 4, it can be concluded that the point, goes to class A, since it has 3 (majority) nearest neighbors from that category.

Fig. 3. Calculation of Euclidean Distance b/w two points [4].



XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

2.2 Advantages of KNN algorithm

- K-NN algorithm is an easy-to-apply algorithm to problems.
- K-NN algorithm is tolerant and resistant to the noise prevailing in the data set used for training.
- It is fast and easy to interpret and effective even if the data set is large enough.

2.3 Disadvantages of KNN algorithm

- Deciding a suitable value for K is a complexity, as it drastically changes the results sometimes.

- Since, the requirement is to calculate the Euclidean type distance between each and every data point belonging to the dataset used for training; it leads to high cost of computation.

3. Long Short-Term Memory (LSTM) algorithm

Due to backpropagation with real-time recurrent learning or time, the error-incorporated signals running rearward in time are likely to disappear or blow up; the temporal shifts of the error incorporated signal to a great extent relies on the weight sizes. In case of blowing up, the weights are quite likely to start oscillating and in case of disappearance, either the time consumed to learn bridging longer time lags is out of bounds, or in the worst case it does not work [5]. As a remedy, the Long Short-Term Memory (LSTM) algorithm, a novel type of recurrent neural network came into existence in 1991, developed by Sepp Hoch Reiter and Jurgen Schmidhuber to outperform the existing systems and overcome the error backpropagation issues discussed above. The primary version of this long short-term memory algorithm only consisted of cells, input, and output gates. This algorithm is capable of bridging time breaks in excess of steps even when the sequences being used for input are incompressible or noisy in nature while preventing losses of short time break abilities. Long Short-Term Memory, designed by Hoch Reiter & Schmidhuber is a special case recurrent neural network (RNN) that is well equipped to handle long term dependencies by default. In LSTM algorithm, the input of a current step is the output of the previous step, thereby solving the issues of long-term dependencies of RNN where the RNN give precise predictions on recent information but are incapable of predicting data stored in long term memory. However, with the increasing gap length, the efficiency of RNN decreases. Some of the major applications of LSTM are captioning images, generation of handwriting chatbots answering questions, and various others [6]

3.1. Structure of LSTM

LSTM structure has been depicted below, consisting of four neural networks and various memory blocks known as the cells. The gates perform memory manipulations on the data stored in cells. The gates are of three types.

3.1.1. Forget gate

The information that is not needed anymore is removed from the cell using the forget gate. The input at a specific time i.e. x_t and the output of the previous cell h_{t-1} are multiplied by the weighted matrices and further addition of bias. To get a binary output the resultant is passes through an activation function. The information in the cell state is then retained if the output is '1' and erased if the output is '0'.

3.1.2. Input gate

It performs the function of adding vital information in a cell state. The information is processed through a sigmoid function and the values to be retained are filtered. Next step involves vector creation using the function tanh which gives an output ranging from -1 to $+1$, containing all possible values from h_{t-1} and x_t. Lastly, vector values and sigmoid function filtered results are multiplied to derive useful results.

3.1.3. Output gate

It declares the output based on data stored in the current cell state. Firstly, a vector is created with the help of the function tanh for cell values. Next step involves regulation of the information using sigmoid function and filtering of the values that are to be retained. Lastly, the product of the regulated values and vector values is sent as an output, which acts as an input for next cell [6].

3.2. LSTM working

The foremost stage requires a decision regarding the removal of unnecessary information from the cell state. Such decisions are resolved by the 'forget gate layer', which is one of the simoid layer [7].decision making, x_t and h_{t-1} are considered and the outcomes for all the numbers belonging to the cell C_{t-1}, could be any number ranging from 0 to 1. In case, the output is '1', it indicates that the information has to be saved, where as a '0' indicates that the information needs to be discarded.

References

[1] Fix E., Hodges J.L. Discriminatory Analysis, Nonparametric Discrimination: Consistency Properties: Technical Report 4

[2]Bansal M., Singh H.
The genre of applications requiring the use of IoT in Day-to-Day Life

[3] Hellman M.E.
The nearest neighbor classification rule with a reject option
IEEE Trans. Syst. Man Cybern., 3 (1970), pp. 179-185

[4]https://images.app.goo.gl/WQbK8Ak4KaFzQs6r9.
Google Scholar

[5] Bansal M., Oberoi N., Sameer M. IoT In online banking
J. Ubiquit. Comput. Commun. Technol.
(UCCT), 2 (4) (2020), pp. 219-222

[6] Bansal M., Sirpal V., Choudhary M.K.Advancing e-government using internet of things Shakya S., *et al.* (Eds.)

[7] Bansal M., Sirpal V., Choudhary M.K. Advancing e-government using internet of things Shakya S., *et al.* (Eds.),