Comparision between SVM and Neural Network

Nitesh Sukhwani

Dept. of Data Science and Analytics

Central University of Rajasthan

Kishangarh,Rajasthan

niteshsukhwani97@gmail.com

***Abstract*—This study concentrated on the comparison of SVM (Support Vector Machine) and FNN (Feedforward Neural Network) on the Titanic dataset. The experiments are performed in the same condition for both models. The comparison of models is done on the basis of accuracy in train and validation set also the cross validation score.**

Keywords*—*SVM, FNN, Cross Validation, Accuracy

I. Introduction

Support Vector Machine and Neural Networks are two different machine learning approach for classification and regression problem with very interesting properties. Although both models have different background there can be establish a direct correspondence between them as both SVM and NN induce the output function which is expressed in terms of simple linear function.

This work focuses on the comparison of SVMs and the FNNs(feed forward Neural Network). An experimental study on Titanic data set for classification problems is presented. All the tested models have in common the property that their hidden-layer weights are a subset of the data. The goal is finding out whether FNNs are competitive with SVMs when both models are restricted to use similar hidden layer weights. For SVMs, sseveral kernel function are tested, namely, the standard one linear , poly and rbf. The experiments were performed in the same conditions for all the models. Both the model tested with the same training and test data sets. The model selection process was as similar as possible, taking into account that different methods need different parameters.

II. Theory

In this section I will discuss the theoretical aspect of both the support vector machine as well as the feed forward neural network.

A. Support Vector Machine(SVM)

Support Vector Machine (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiate the two classes. Formally a Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labelled training data (*supervised learning*), the algorithm outputs an optimal hyperplane which categorizes new examples. In two dimensional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side. There are mainly four parameters of any SVM and those are as following

a. Kernel: The learning of the hyperplane in linear SVM is done by transforming the problem using some linear algebra. This is where the kernel plays role. For linear kernel prediction of new input using the dot product between the input X and each support vector X_i is calculated as follows

$$f(x) = b(0) + sum\left(a_i * (x, x_i)\right)$$

For polynomial kernel this can be written as

$$k(x, x_i) = 1 + sum(x * x_i)^d$$

Where

$$k(x, x_i) \ = \ exp(-\gamma * \ sum(x - x_i^2))$$

This is known as kernel trick.

b. Regularization: This tells SVM about how much to avoid misclassification. For large values of C, the optimization will choose a smaller-margin hyperplane if that hyperplane does a better job of getting all the training points classified correctly. Conversely, a very small value of C will cause the optimizer to look for a larger margin separating hyperplane, even if that hyperplane misclassifies more points.

c. Gamma: The gamma parameter defines how far the influence of a single training example reaches, with low values meaning ‘far’ and high values meaning ‘close’. In other words, with low gamma, points far away from plausible separation line are considered in calculation for the separation line. Whereas high gamma means the points close to plausible line are considered in calculation.

d. Margin: A margin is a separation of line to the closest class points. A good margin is one where this separation is larger for both the classes. A good margin allows the points to be in their respective classes without crossing to other class.

B. Feed Forward Neural Network(FNN)

a neural network emulates the human brain. Brains cells, or neurons, are connected via synapses. This is abstracted as a graph of nodes (neurons) connected by weighted edges (synapses). A neural network has input and output neurons, which are connected by weighted synapses. The weights affect how much of the forward propagation goes through the neural network. The weights can then be changed during the back propagation — this is the part where the neural network is now learning. This process of forward propagation and backward propagation is conducted iteratively on every piece of data in a training data set. The greater the size of the data set and the greater the variety of data set that there is, the more that the neural network will learn, and the better that the neural network will get at predicting outputs. These are the component of a Neural Network

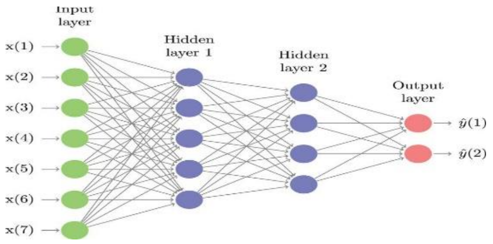
a. Neuron: A neural network is a graph of neurons. A neuron has inputs and outputs. Similarly, a neural network has inputs and outputs. The inputs and outputs of a neural network are represented by input neurons and output neurons. Input neurons have no predecessor neurons, but do have an output. Similarly, an output neuron has no successor neuron, but does have inputs.

b. Weights: A neural network consists of connections, each connection transferring the output of a neuron to the input of another neuron. Each connection is assigned a weight.

c. Propagation Rule: The propagation function computes the input of a neuron from the outputs of predecessor neurons. The propagation function is leveraged during the forward propagation stage of training.

d. Learning Rule: The learning rule is a function that modifies the weights of the connections. This serves to produce a favoured output for a given input for the neural network. The learning rule is leveraged during the backward propagation stage of training.

A Deep Neural Network simply has more layers than smaller Neural Networks. A smaller Neural Network might have 1–3 layers of neurons. However, a Deep Neural Network (DNN) has more than a few layers of neurons. A DNN might have 20 or 1,000 layers of neurons.



III. Dataset

The data used in this paper was titanic dataset which is freely available at kaggel.com. The dataset comprises of total 891 instance and 11 features and 1 target variable. The target variable is distributed in 40-60 ratio i.e. 60 percent of the instance are labelled with value 0 and remaining labelled with value 1. The attributes of dataset are as follows

Attribute Name	Data Type	Description
PassengerId	Int	Unique ID assigned
Survived	Int	Whether passenger survived
Pclass	Int	Ticket class
Name	Object	Name of the passenger
Sex	Object	Gender of the passenger
Age	Float	Age of the passenger
Sibsp	Int	No. of sibling /spouse
Parch	Int	No. of family member
Ticket	Object	Number on ticket
Fare	Float	Fare of the ticket
Cabin	Object	Cabin name
Embarked	Object	Port of Embarkation

Table 1: Dataset Description

The dataset is divided into 70-30 ratio for train and validation set.

IV. Preprocessing of Dataset

In this section we discuss the main pre processing steps involved before fitting any model. Firstly, we remove the features PassengerId, Cabin, Name and Ticket as these won't help in anywhere in model building. After removal of these unnecessary features we have make all other feature in model understandable format for that we have to encode each object feature to categorical type.

Now the important task is to impute the missing value of each column for that we have used MICE(Multivariate Imputation using Chained Equation) a package built in R which treat the column with missing value as target variable and regressed it over all other variable and impute the value based on the prediction given by regressor. MICE use linear regression for continuous and logistic regression for categorical features. Finally we scale the to Normal (0,1).

V. Results

As the main aim of our study is to show the difference between the SVM and three-layer neural network we have used these two models on titanic dataset. The results of the SVM are with rbf kernel and gamma =0.09 and for neural network our hidden layers consisting of 100 neurons. We have used Adam optimiser with activation function tanh. The accuracy score on train and validation dataset and also cross validation score with k =10 are given in table 2.

Model Name	Train	Test	CV
SVM	83.53	82.06	83.15
Neural Network	83.71	81.72	79.80

Table 2: Accuracy score of SVM and FNN

VI. Conclusion

The experiment in the study be a comparison between the SVM and FNN. In our model SVM obtain the slightly higher accuracy than FNN also computation time is less for SVM but this may dependent on implementation and code optimizations. Therefore, we conclude SVM are better than FNN in these types of smaller datasets.

REFERENCES

[1] Enrique romero and Daniel Toppo ‘Comparing Support Vector Machines and Feedforward Neural Networks With Similar Hidden-Layer Weights’.

[2] https://stackabuse.com/implementing-svm-and-kernel-svm-with-pythons-scikit-learn/

[3] https://towardsdatascience.com/predicting-the-survival-of-titanic-passengers-30870ccc7e8