INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

IJARIIT

ISSN: 2454-132X
Impact Factor: 6.078
(Volume 7, Issue 4 - V7I4-1174)
Available online at: https://www.ijariit.com

Improving the accuracy of Neural Networks through Ensemble Techniques

Monalika Padma Reddy
monalika.6671@gmail.com
Visvesvaraya Technological University, Belgaum, Karnataka

Deeksha A.
deekshaarun1523@gmail.com
Visvesvaraya Technological University, Belgaum, Karnataka

ABSTRACT

Neural networks usually suffer from model performance problems due to high variance. Ensemble techniques refer to the method of combining multiple models resulting in improving the overall performance of the model. The overall performance can be improved by combining the predictions from multiple models having a good accuracy instead of highly tuning the models, to reduce the variance of predictions and reduce the error due to generalization. In this paper, the different types of ensemble techniques such as stacking ensemble, horizontal voting, and weight average ensemble are being discussed which can be used to improve the model accuracy. The numerous disadvantages of hyperparameter tuning like the wrong choice of parameters, overfitting, and other inefficient optimizing strategies can be overcome using ensemble techniques. The ensemble techniques have numerous advantages such as improving the model performances, reduction of model variance, and so on. Accuracy is improved because results are obtained using the mean predictions of the number of sub-models and the performance is improved with the learning weights of each of the models.

Keywords— *Ensemble techniques, Horizontal Voting, Hyperparameters, Neural Networks, Stacked generalization ensemble, Weight Average Ensemble*

1. INTRODUCTION

Neural networks usually suffer from model performance problems due to high variance. This is because they are highly flexible. The weights are calculated and updated using algorithms like gradient descent and stochastic gradient descent. This makes the network highly sensitive to the training data. There are chances of having a different set of weights each time the model is trained, which may result in different predictions. The accuracy of the neural network model can be improved by hyperparameter tuning. Hyperparameter tuning can be done by varying the number of hidden layers, activation functions, weight initializations, and other such techniques and choosing the best set of hyperparameters. This method has a lot of disadvantages because trying out several different combinations of hyperparameters is a tedious task and hyperparameter tuning may result in overfitting and so on.

The Ensemble technique is a method of combining multiple models and generating a model, which results in improving the accuracy, stability, and overall performance of the model. Ensemble techniques are usually used to increase the model performance by reducing the variance of predictions and reducing the error due to generalization. The multiple models that are being combined are called base learners.

Every machine learning algorithm usually suffers from statistical problems, computational problems, and representation problems. The errors in any machine learning models can be described as shown in Equation(1)

$$d_{f,\theta}(y,t) = \text{Bias}_{\theta} + \text{Variance}_{f} + \text{Noise}_{t} \tag{1}$$

These errors can be overcome by using ensemble techniques thereby improving the model performance. An ensemble model usually consists of several models that are generated from the training data. The ensemble techniques are well known for their ability to boost weak learners.

2. ENSEMBLE TECHNIQUE

Ensemble techniques are also called learning multiple classifier tech or committee-based learning because the model will be trained on multiple hypotheses to solve the same problem. Each of the models makes a significant contribution to the final prediction and the inaccuracy of each model is offset by the contribution of the other models.

The accuracy is improved because the results are obtained using the mean predictions of the number of base learners and the performance is improved with the learning weights of each of the models. The number of models that are used in an ensemble is small because of two reasons - computational expenses in training the models and performance decrease when more ensemble models are used.

Ensemble techniques are meta-algorithms that combine several machine learning models into a single model to improve predictions or decrease variance and bias. They can be divided into two types:

1. Sequential ensemble techniques
2. Parallel ensemble techniques

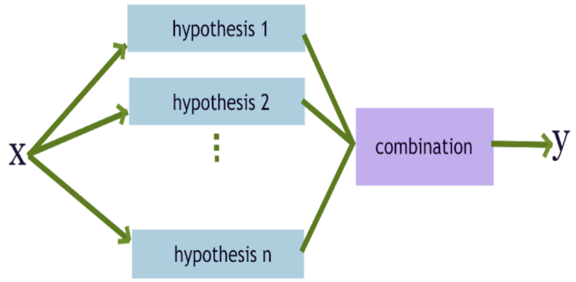

Fig.1. A general ensemble architecture

In sequential ensemble techniques, the base learners are generated sequentially, and the dependence between the base learners is exploited. The overall performance of the model can be increased by weighing mislabelled examples with higher weight. In parallel ensemble techniques, the base models are generated parallelly, and the independence of the models is exploited because the error can be dramatically used by averaging.

Even though ensemble methods have great complexity and computational cost, an ensemble model will be better than a single model for the following reasons:

(a) Performance - Better and more accurate predictions can be made by an ensemble model than any single contributing model.
(b) Robustness - An ensemble model dispersion or a spread of predictions thereby improving the model performance significantly.

2.1. Advantages of ensemble techniques

Ensemble techniques are used to enhance the perf of the models. It involves the construction of a set or a group of base learners and aggregating the output of all the classifiers for prediction. The advantages of ensemble techniques are as follows:

(a) Ensemble techniques can capture both the linear as well as non-linear relationships in the data. Two or more models can be used to form an ensemble model.
(b) The aggregate result of multiple models is less noisy than each of the individual models which leads to model stability and robustness.
(c) The ensemble models will provide better accuracy because multiple models will be combined and produce better predictions when compared to a single model.

2.2. Applications of ensembles techniques

The applications of ensemble techniques are as follows:

(a) Ensemble techniques can be used for the evaluation of the relationships between responses in conventional statistical models and explanatory variables.
(b) Ensemble techniques can be used to capture the selection process better and the probability in each group can be estimated with less bias.
(c) Ensemble techniques can be used to implement covariance adjustments in multiple regression and other related procedures.
(d) Ensemble techniques can be used for a more conventional model building as an overall diagnostic procedure.

3. NEURAL NETWORK ENSEMBLE

Ensemble technique is a method where multiple models are combined to provide an output. This is because multiple models when combined perform better than the individual models. The neural network ensemble [1] is an ensemble technique where multiple neural network models are combined to provide an output. It not only improves the performance of the neural network by dampening the inherent sensitivity to noise, but also by combining uncorrelated and qualitatively different solutions. There are two steps involved in a neural network ensemble. They are:

(a) Training several neural networks– Training neural network starts with the random initialization of weights to each node in the neural network. The neural networks can be trained either using a supervised or non-supervised training approach. In supervised training, both the inputs and the outputs are provided to the neural network, which processes the inputs, compares it obtained results against the actual outputs.
The errors obtained are then propagated through the entire network, thereby adjusting the weights. This is a continuous process where the weights are tweaked. In unsupervised training, the model is provided with inputs only. The model decides which input features it will use to group the data.
(b) Combining the predictions obtained by component neural networks- Combining the predictions involves the process of using each of the trained models to make predictions before obtaining the final prediction. The models add a bias, that counters the variance of each neural network model.
The results obtained are less sensitive to the choice of the training scheme, specifics of the training data, and so on.

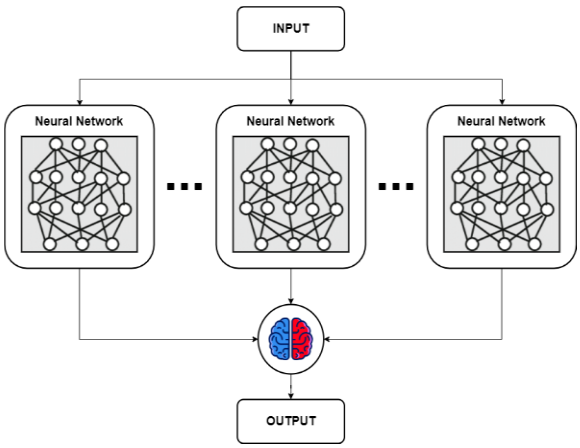

Fig. 2. Neural Network Ensemble

3.1 Neural Network Ensemble Techniques

Neural network models are non-linear and have high variance. Ensemble techniques combine the predictions made by multiple neural network models to reduce generalization error and variance of predictions.

3.1.1 Stacked Generalization Ensemble: Stacking ensemble models are a further generalization of the weighted average ensemble where the linear weighted sum model that is used to combine the predictions of sub-models is replaced with any other learning algorithm. It works by deducing the generalizer's bias for the learning set. Stacked generalization [3] makes use of a meta-learner. A meta-learner is a second-level machine algorithm that uses an optimal combination of base learners to learn.

The algorithm takes the output of several sub-models as the input to make a better output prediction by attempting to learn to combine the best input predictions. The base-level models are trained on the complete training set and the meta-models are trained on the output features of the base-level models.

The stacking model consists of two or more models called level-0 models and meta-models combine the predictions of the base models and are known as level-1 models. The output of the base models which are given as an input to the meta-models can be real values, probability, or probability-like values or class labels. Fig.3. represents stacked generalization ensemble.

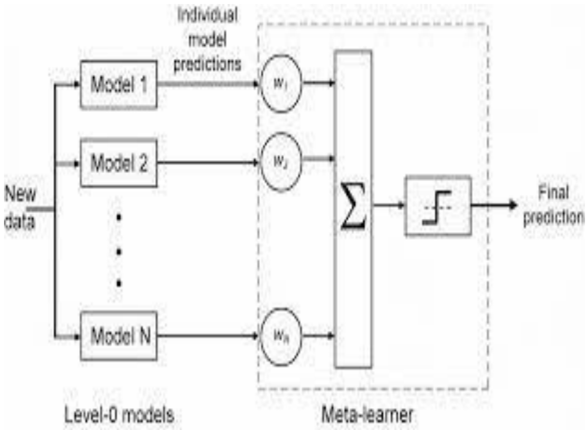

Fig. 3. Stacked generalization ensemble

Let x_i be the set of observations and y_i be the set of labels for each observation

Let D = {(x_i,y_i)} be the training dataset given to the base models.

Let B be the base models and M be the meta models

Algorithm : Stacking generalization ensemble

Input : Training dataset D = {(x_i,y_i)}
Output : prediction – final prediction obtained by the stacked generalization ensemble model

1. **Step 1 :** Train the base-level models
2. **for each** t in T
   Train a base model $B_t \in$ B based on D
3. end **for**
4. **Step 2 :** Train the meta-level models by
5. **for each** t in T
   Train a meta-model $M_t \in$ M by giving the base models B as the input
6. end **for**
7. **Step 3 :** Obtain prediction by using the obtained stacking ensemble model
8. **Step 4 : return** prediction

3.1.2 Horizontal Voting Ensemble: A horizontal voting ensemble [4] is a method that involves using a collection of models that are saved over continuous training epochs are saved as an ensemble. This results in more stability and better performance than choosing a single final model randomly. It provides a way to improve the average performance of the model and reduce variance using a single training run.

Neural networks perform well on predictive modelling problems. They may suffer from model performance due to high variance which chooses models to be used as a final model risky. This is because there is no clear indication as to which model performs better towards the end of training. The horizontal voting ensemble can be used to address this. Fig.4. represents the horizontal voting ensemble.

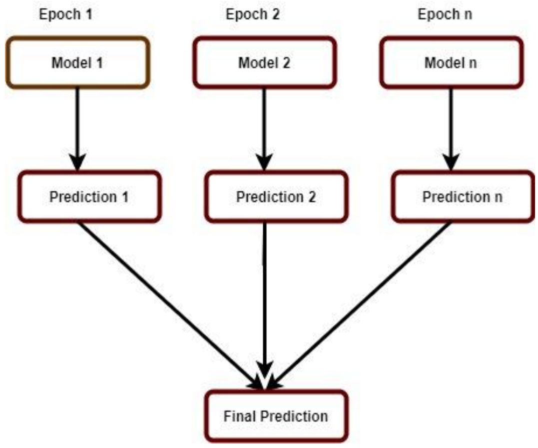

Fig. 4. Horizontal voting ensemble

Let x_i be the set of observations and y_i be the set of labels for each observation.
Let D = {(x_i,y_i)} be the training dataset, T = {(x_i',y_i')} be the test dataset
Let E be the number of epochs
Let M be the set of all models trained and saved over contiguous training epochs
Let 'predict' be the function for returning the prediction a model m, where m ∈ M
Let P be a list containing all the predictions made by models in M on test set T

Algorithm : Horizontal voting ensemble

Input : Training dataset D = {(x_i,y_i)}
Output : prediction – final prediction obtained by the horizontal voting ensemble model

1. **Step 1 :** Train and save the models over contiguous training epochs
2. **for each** e in E
   Train a model m ∈ M based on D
   Save the model
3. End for
4. **Step 2 :** Load all the trained models in M
5. **Step 3 :** Obtain the predictions from all trained models in M and append it to a list 'P'
6. **for each** m in M
   pred ← m(predict(T))
   P ← **append**(pred)
7. end **for**
8. **Step 4 :** Obtain the final prediction
9. prediction ← value having the most frequency in P (for classification)
10. **Step 5 : return** prediction

Horizontal voting ensemble provides an ideal method in cases where a given model needs vast computational resources for training and selecting a final model is challenging because of the high variance in training which is caused due to the use of a relatively small dataset for training. It involves using multiple models, before the end of training from continuous blocks of epochs in an ensemble to make the predictions. It reduces instability.

3.1.3 Weight Average Ensemble: Weight average ensemble [5] is an ensemble technique that allows multiple models to contribute to the prediction, where the contribution of each model is directly proportional to its estimated performance. Model weights are small positive values where the sum of all the weights will be equal to one. This allows the weights to be indicated as a percentage of each model's expected performance. Finding the weight of the ensemble members can be done by using approaches such as grid search and optimization.

The accuracy of each model will be used as its corresponding weight. Fig.5. represents the working of the weight average ensemble.

Weight average ensemble is an extension of a model averaging ensemble where the contribution of each model is weighed by model performance Model averaging is an ensemble technique where each model contributes to the final prediction equally.

For instance, in the case of a regression problem, the final prediction will be the average of all the predictions of all the models. In the case of a classification problem, the class label will be calculated as the argmax of the sum of the probabilities for each class. But the disadvantage of this method is that it requires each model to make an equal contribution to the final prediction and hence, a weight average ensemble is used.

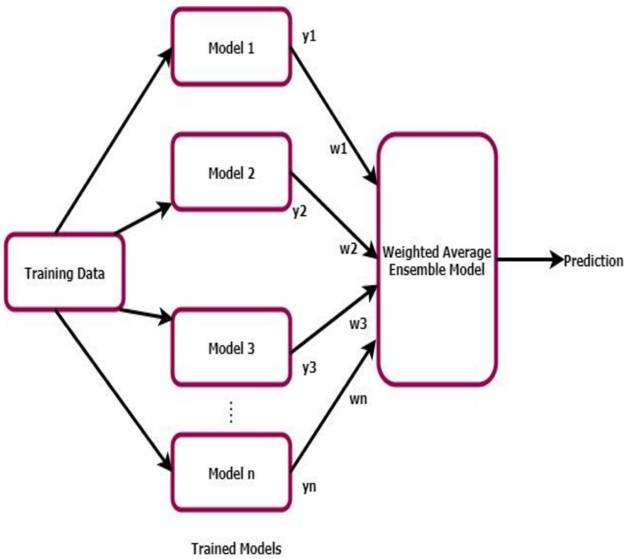

Fig. 5. Weight average ensemble

For each observation, let x_i be the set of observations and y_i be the set of labels

Let D = {(x_i,y_i)} be the training dataset

Let M be the set of all models trained on the same dataset D

Let 'accuracy' be the function for returning the accuracy of a model m, where m ∈ M.

Let modelAccuracy be a list containing all the accuracy of each model will be used as its corresponding weight.

Algorithm : Weight average ensemble

Input : Training dataset D = {(x_i,y_i)}

Output : prediction – final prediction obtained by the weight average ensemble model

Step 1 : Train the models on the training set D and save the models
for each e in E
    Train a model m ∈ M based on D
    Save the model
end **for**
Step 2 : Evaluate each trained model in M
for each m in M
    modelAccuracy ← **append**(accuracy(m))
end **for**
Step 3 : Create ensemble model using models m in M and accuracy in modelAccuracy as the corresponding weight of each model
Step 4 : Train the ensemble model
Step 5 : Obtain the final prediction
prediction ← average of all the predictions in P (for regression)
Step 6 : return prediction

4. EXPERIMENTAL RESULTS

The most common machine learning and deep learning datasets are used for testing the ensemble models. Good benchmark results could be achieved without having to perform any hyper-parameter tuning.

The models were tested with Root Mean Squared Error (RMSE), Mean Absolute Error(MAE), and r2 score for regression problems and metrics accuracy, precision, recall, and f1 score for classification problems.

The important observations made are as follows –

1. Neural networks have high variance and are non-linear and this causes certain issues while preparing the model for predictions
2. Ensemble methods reduce the variance of prediction by combing model weights from different models.
3. Ensemble methods reduce the generalization error

(a)

Models	Precision	Recall	F1 Score	Accuracy
Random	0.64	0.63	0.65	0.83
Model 1	0.75	0.55	0.68	0.84
Model 2	0.79	0.62	0.68	0.85
Ensemble	**0.79**	**0.63**	**0.70**	**0.89**

(b)

Models	**RMSE**	**MAE**	**R2 Score**
Model 1	54675.8	35326.6	0.766
Model 2	53753.6	35326.6	0.762
Ensemble	**52425.4**	**35326.6**	**0.789**

(c)

Models	**Accuracy**
Model 1	0.846
Model 2	0.858
Stacked Model	**0.89**

The experimental result on the classification dataset is shown in Table(a). The models were trained on a classification dataset. The models Model 1 and Model 2 were saved using callbacks and have low validation loss and the highest validation accuracy. The accuracy of all the models – Random, Model 1, and Model 2 all have an accuracy score around 83% - 85%. The Fig.6. depicts the line plot showing single model accuracy vs accuracy of ensembles of different sizes with a horizontal voting ensemble [4]. On using the horizontal voting, the accuracy of the ensembled model is 89%. It can be seen that the accuracy increased by 4% which is an excellent improvement when dealing with large datasets. The recall and the precision score improved as well.

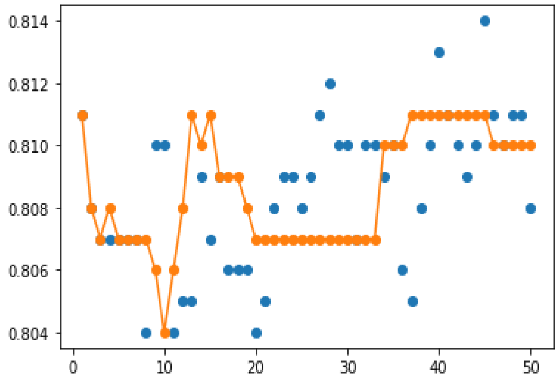

Fig.6. Graph of Single Model Accuracy vs Accuracy of Ensembles of different size With a Horizontal Voting Ensemble

The experimental result on the regression dataset is shown in Table(b). Both the neural network models, Model 1 and Model 2 are saving using the same neural network architecture. During the last 20 epochs, models were saved at every epoch and are used for weight average ensemble [5]. The Fig.7. shows the line plot depicting a single model accuracy and accuracy of ensembles models of increasing size. Blue dots represent the single model accuracy and the orange line shows the accuracy of ensembles models of increasing size. The ensemble model obtained had a high R2 score which was found out to be 0.78.

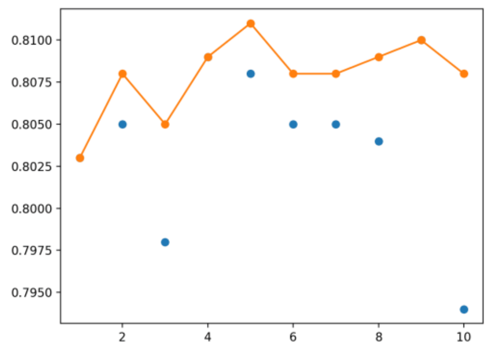

Fig.7. Line Plot Depicting a Single Model Accuracy and Accuracy of Ensembles models of Increasing Size

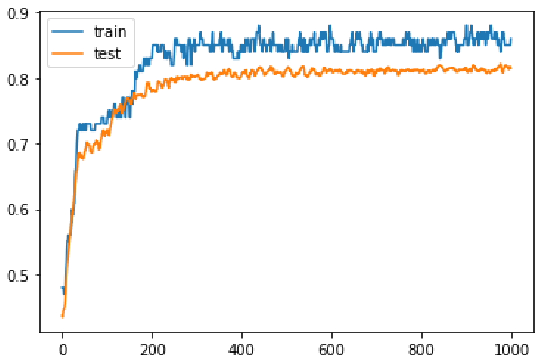

Fig.8. Graph of learning curves of model accuracy on train dataset and test dataset over each training epoch

The experimental result on the classification dataset is shown in Table(c). Fig.8. shows the graph of learning curves of model accuracy on train dataset and test dataset over each training epoch. Model 1 and Model 2 are two different neural network architectures trained on a classification dataset, with an accuracy of 84.6% and 85.5% respectively. On using the stacking generalization ensemble [3], the model accuracy increased to 89%.

5. CONCLUSION

Neural networks are a series of algorithms that mimic the working of the human brain in order to recognize the relationship between large amounts of data. Neural networks suffer from model performance problems due to high variance and it is often a tedious task to build the State of the Art (SOTA) models. There are numerous methods [6] [7] to improve the performance of the neural network models, such as hyperparameter tuning and so on, but they are time-consuming. There are many disadvantages such as the wrong choice of parameters during hyperparameter tuning, overfitting, and other inefficient optimizing strategies. Ensemble techniques have numerous advantages such as improving the model performances, reducing model variance, and can be used to improve the performance of the models.

6. REFERENCES

[1] L. K. Hansen and P. Salamon, "Neural network ensembles," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no. 10, pp. 993-1001, Oct. 1990, doi: 10.1109/34.58871.

[2] Y. Liu, X. Yao, Ensemble learning via negative correlation, Neural Networks, Volume 12, Issue 10,1999, Pages 1399-1404, ISSN 0893-6080, https://doi.org/10.1016/S0893-6080(99)00073-8.

[3] David H. Wolpert, Stacked generalization, Neural Networks, Volume 5, Issue 2, 1992, Pages241-259, ISSN 0893-6080 https://doi.org/10.1016/S0893-6080(05)80023-1.

[4] Jingjing Xie, Bing Xu, Zhang Chuang, Horizontal and Vertical Ensemble with Deep Representation for Classification, June 2013, arXiv:1306.2759

[5] Michael P. Perrone and Leon N. Cooper (1995), When networks disagree: Ensemble methods for hybrid neural networks https://doi.org/10.1142/9789812795885_0025

[6] MacKay D.J.C. (1995) Developments in Probabilistic Modelling with Neural Networks —Ensemble Learning. In: Kappen B., Gielen S. (eds) Neural Networks: Artificial Intelligence and Industrial Applications. Springer, London. https://doi.org/10.1007/978-1-4471-3087-1_37

[7] Polikar R. (2012) Ensemble Learning. In: Zhang C., Ma Y. (eds) Ensemble Machine Learning. Springer, Boston, MA. https://doi.org/10.1007/978- 1-4419-9326-7_1

