4.1 COMBINING MULTIPLE LEARNERS

Ensemble learning is one of the most powerful machine learning techniques that use the combined output of two or more models/weak learners and solve a particular computational intelligence problem. E.g., a Random Forest algorithm is an ensemble of various decision trees combined.

Ensemble learning is primarily used to improve the model performance, such as classification, prediction, function approximation, etc. When designing a learning machine, we generally make some choices like parameters of machine, training data, representation, etc. This implies some sort of variance in performance. For example, in a classification setting, we can use a parametric classifier or in a multilayer perceptron, we should also decide on the number of hidden units.

Each learning algorithm dictates a certain model that comes with a set of assumptions. This inductive bias leads to error if the assumptions do not hold for the data.

• Different learning algorithms have different accuracies. The no free lunch theorem asserts that no single learning algorithm always achieves the best performance in any domain. They can be combined **to attain higher accuracy**

- **Data fusion** is the process of fusing multiple records representing the same real-world object into a single, consistent, and clean representation Fusion of data for improving prediction accuracy and reliability is an important problem in machine learning.

• **Combining different models** is done to improve the performance of deep learning models. Building a new model by combination requires less time, data, and computational resources. The most common method to combine models is by averaging multiple models, where taking a weighted average improves the accuracy.

1. Generating Diverse Learners:

Different Algorithms: We can use different learning algorithms to train different base-learners. Different algorithms make different assumptions about the data and lead to different classifiers.

Different Hyper-parameters: We can use the same learning algorithm but use it with different hyper-parameters.

Different Input Representations: Different representations make different characteristics explicit allowing better identification.

Different Training Sets: Another possibility is to train different base-learners by different subsets of the training set.

4.1.1 Model Combination Schemes

Different methods are used for generating final output for multiple base learners are Multiexpert and multistage combination.

1. **Multiexpert combination**

Multiexpert combination methods have base-learners that work in parallel.

a) **Global approach** (learner fusion) given an input, all base-learners generate an output and all these outputs are used, such as voting and stacking

b) **Local approach** (leamer selection) in mixture of experts, there is a gating model, which looks at the input and chooses one (or very few) of the learners as responsible for generating the output.

2. Multistage combination: Multistage combination methods use a serial approach where the next multistage combination base-learner is trained with or tested on only the instances where the previous base-learners are not accurate enough.

- Let's assume that we want to construct a function that maps inputs to outputs from a set of known N_{train}input-output pairs.

$$D_{train} = \{(x_i, y_i)\}_{i=1}^{N_{train}}$$

where x, Є X is a D dimensional feature input vector. y_iЄ Y is the output.

Classification: When the output takes values in a discrete set of class labels Y = {c1;c2;…cx}, where K is the number of different classes. Regression consists in predicting continuous ordered outputs, Y=R

4.1.2 Voting

- The simplest way to combine multiple classifiers is by voting, which corresponds to taking a linear combination of the learners. Voting is an ensemble machine learning algorithm.

• For regression, a voting ensemble involves making a prediction that is the average of multiple other regression models.

A Voting Classifier is a machine learning model that trains on an ensemble of numerous models and predicts an output (class) based on their highest probability of chosen class as the output.

- It simply aggregates the findings of each classifier passed into Voting Classifier and predicts the output class based on the highest majority of voting.
- The idea is instead of creating separate dedicated models and finding the accuracy for each them, we create a single model which trains by these models and predicts output based on their combined majority of voting for each output class.

Voting Classifier supports two types of votings.

Hard Voting: In hard voting, the predicted output class is a class with the highest majority of votes i.e the class which had the highest probability of being predicted by each of the classifiers. Suppose three classifiers predicted the *output class(A, A, B),* so here the majority predicted *A* as output. Hence *A* will be the final prediction.

Soft Voting: In soft voting, the output class is the prediction based on the average of probability given to that class.

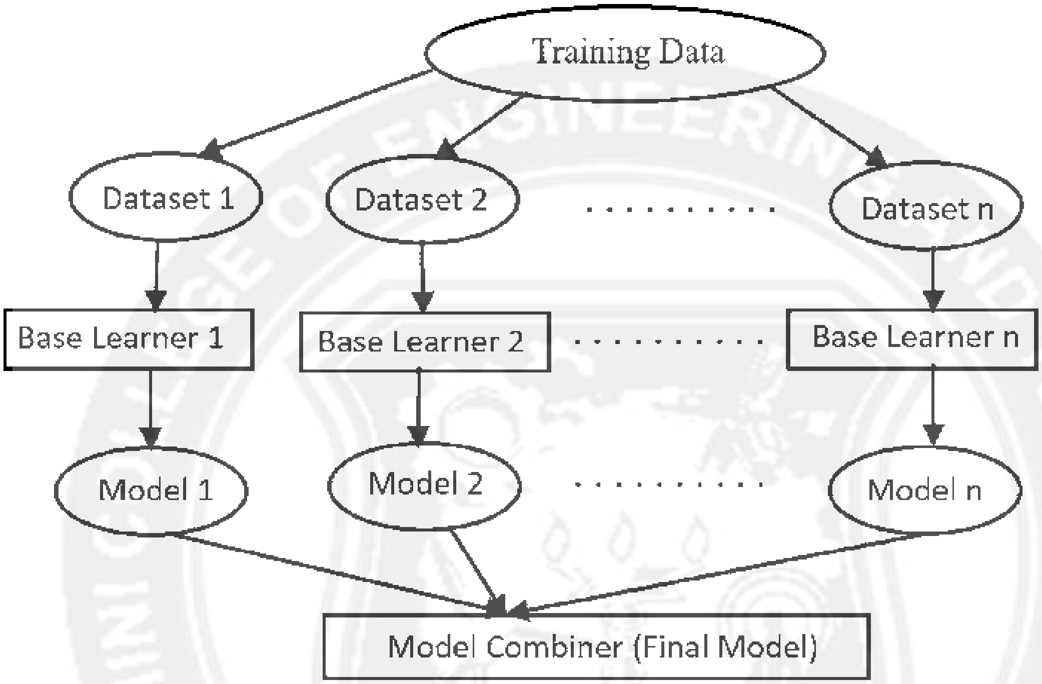
- Suppose given some input to three models, the prediction probability for class *A = (0.30, 0.47, 0.53)* and *B = (0.20, 0.32, 0.40)*. So the average for class *A is 0.4333* and *B is 0.3067*, the winner is clearly class *A* because it had the highest probability averaged by each classifier.

In this methods, the first step is to create multiple classification/regression models using some training dataset. Each base model can be created using different splits of the same training dataset and same algorithm, or using the same dataset with different algorithms, or any other method.

- Learn multiple alternative definitions of a concept using different training data or different learning algorithms. It combines decisions of multiple definitions,

e.gusing weighted voting

Figure shows general idea of Base-learners with model combiner



When combining multiple independent and diverse decisions each of which is at least more accurate than random guessing, random errors cancel each other out, and correct decisions are reinforced. Human ensembles are demonstrably better