

# Ensemble Learning

Wenbin Wang

Advisor: Prof. Jinhua Zhao & Dr. Xinyu Chen

School of Information Science and Technology  
ShanghaiTech University, Shanghai, China

MIT-UF-NEU 2025 Summer Research Camp

July 22th, 2025

# Outline

- Introduction
- Types & Popular Algorithms
- Challenges & Current Research Progress
- Future Directions
- Summary

# Outline

- Introduction
- Types & Popular Algorithms
- Challenges & Current Research Progress
- Future Directions
- Summary

## Reference books and materials

- [http://www.scholarpedia.org/article/Ensemble\\_learning](http://www.scholarpedia.org/article/Ensemble_learning)
- [http://en.wikipedia.org/wiki/Ensemble\\_learning](http://en.wikipedia.org/wiki/Ensemble_learning)
- <http://en.wikipedia.org/wiki/Adaboost>
- Rudin video: [http://videlectures.net/mlss05us\\_rudin\\_da/](http://videlectures.net/mlss05us_rudin_da/)

# What is Ensemble Learning?

Ensemble learning is a machine learning technique that combines multiple models to create a more accurate and robust model than any individual model could achieve on its own. It leverages multiple machine learning algorithms collectively to address classification or regression problems. This is done by training several models on the same/independent datasets and then aggregating their predictions to make a final decision.

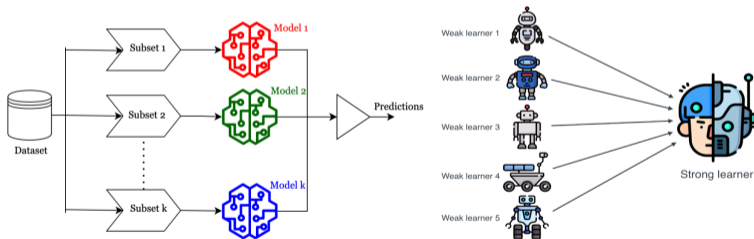


Figure 1: Ensemble Learning<sup>1</sup>

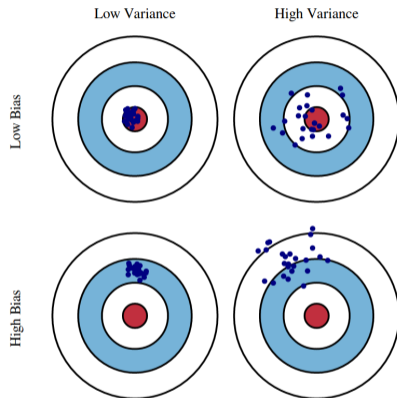
<sup>1</sup><https://images.app.goo.gl/qKHd7LhBxrsUGvhG9> & <https://images.app.goo.gl/sUYxZxKfiC7zzGVE7>

# Why Do We Need Ensemble Learning?

## Rationale:

- **No Free Lunch Theorem:** There is no algorithm that is always the most accurate  
[http://en.wikipedia.org/wiki/No\\_free\\_lunch\\_in\\_search\\_and\\_optimization](http://en.wikipedia.org/wiki/No_free_lunch_in_search_and_optimization)
- **Generate a group of base-learners which when combined has higher accuracy**
- **Each algorithm makes assumptions which might be or not be valid for the problem at hand or not.**
- **Different learners use different**
  - **Algorithms**
  - **Hyperparameters**
  - **Representations (Modalities)**
  - **Training sets**
  - **Subproblems**

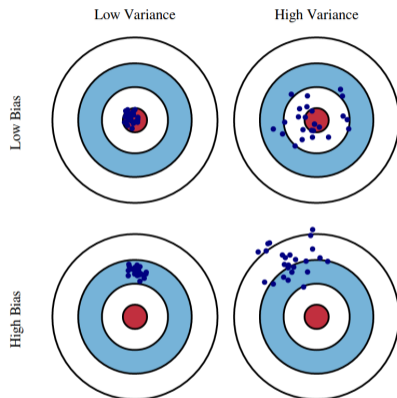
# No Free Lunch Theorem



Bias: Quantify how much on an average are the predicted value different from the actual value<sup>2</sup>.

<sup>2</sup>Image source: G. M. Tina, C. Ventura, S. Ferlito, and S. De Vito, A State-of-Art-Review on Machine-Learning Based Methods for PV, Applied Sciences, vol. 11, no. 16, p. 7550, Aug. 2021, doi: 10.3390/app11167550.

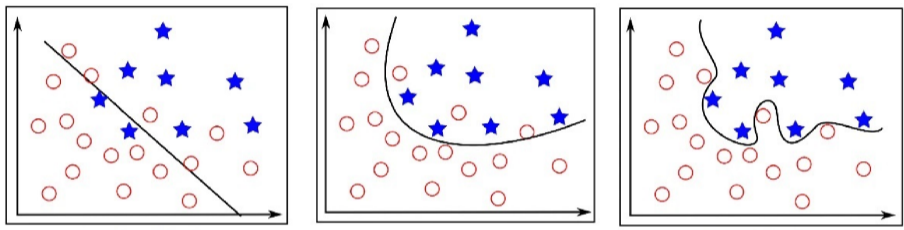
# No Free Lunch Theorem



Variance: Quantifies how much of the predictions made on the same observation different from each other. High variance can cause an algorithm to model the random noise in the training data, rather than the intended outputs.



# No Free Lunch Theorem



High Bias (Under-fitting)

Low Bias, Low Variance

High Variance (Over-fitting)

Image source: <https://vitalflux.com/bagging-classifier-python-code-example/>

# General Idea

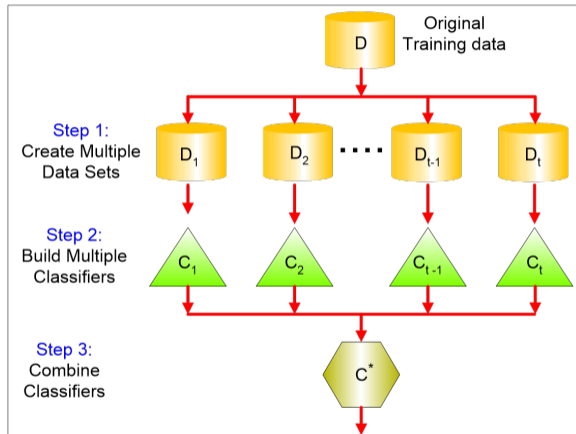


Figure 2: Ensemble Learning

# Why does it work?

- Suppose there are 25 base classifiers
  - Each classifier has error rate,  $\epsilon = 0.35$
  - Assume classifiers are independent
  - Probability that the ensemble classifier makes a wrong prediction:

$$\sum_{i=13}^{25} \binom{25}{i} \epsilon^i (1 - \epsilon)^{25-i} = 0.06$$

## Why are Ensemble Successful?

- **Bayesian perspective:**

$$P(C_i | x) = \sum_{\text{all models } M_j} P(C_i | x, M_j) P(M_j)$$

- If  $d_j$  are independent

$$\text{Var}(y) = \text{Var}\left(\sum_j \frac{1}{L} d_j\right) = \frac{1}{L^2} \text{Var}\left(\sum_j d_j\right) = \frac{1}{L} \text{Var}(d_j)$$

Bias does not change, variance decreases by  $L$

- If dependent, error increases with positive correlation

$$\text{Var}(y) = \frac{1}{L^2} \text{Var}\left(\sum_j d_j\right) = \frac{1}{L^2} \left[ \sum_j \text{Var}(d_j) + 2 \sum_j \sum_{i < j} \text{Cov}(d_i, d_j) \right]$$

# The Advantages of Ensemble Learning

Ensemble learning offers numerous advantages. Here, we will examine its most compelling benefits and explain why its such a powerful tool.

- **Reducing Overfitting**
- **Improving Accuracy**
- **Handling Noisy Data**
- **Handling Imbalanced Data**
- **Handling Large Datasets**
- **Versatility**

# Outline

- Introduction
- Types & Popular Algorithms
- Challenges & Current Research Progress
- Future Directions
- Summary

# Types & Popular Algorithms

There are different types of Ensemble Learning techniques which differ mainly by the type of models used (homogeneous or heterogeneous models), the data sampling (with or without replacement, k-fold, etc.), and the decision function (voting, average, meta model, etc).

Therefore, Ensemble Learning techniques can be classified as:

- **Bagging(Bootstrap Aggregation)**
- **Boosting**
  - **AdaBoost**
  - **Gradient Boosting**
  - **XGBoost**
- **Stacking**
- **Voting**
- **Other ensemble learning approaches**
  - **Bucket of Models**
  - **Bayesian Model Averaging**

# Bagging

- Use bootstrapping to generate  $L$  training sets and train one base-learner with each [1]
- Use voting (Average or median with regression)
- Unstable algorithms profit from bagging
- Example:

- Sampling with replacement

Original Data	1	2	3	4	5	6	7	8	9	10
Bagging (Round 1)	7	8	10	8	2	5	10	10	5	9
Bagging (Round 2)	1	4	9	1	2	3	2	7	3	2
Bagging (Round 3)	1	8	5	10	5	5	9	6	3	7

- Build a classifier on each bootstrap sample
- Each sample has probability  $(1 - \frac{1}{n}) \cdot n$  of being selected



# Bagging



# Boosting

- An iterative procedure to adaptively change the distribution of training data by focusing more on previously misclassified records
  - Initially, all  $N$  records are assigned equal weights
  - Unlike bagging, weights may change at the end of the boosting round
- Records that are wrongly classified will have their weights increased
- Records that are classified correctly will have their weights decreased

- Example:

Original Data	1	2	3	4	5	6	7	8	9	10
Boosting (Round 1)	7	3	2	8	7	9	4	10	6	3
Boosting (Round 2)	5	4	9	4	2	5	1	7	4	2
Boosting (Round 3)	4	4	8	10	4	5	4	6	3	4

- Example 4 is hard to classify
- Its weight is increased, therefore it is more likely to be chosen again in subsequent rounds

# AdaBoost

---

## Algorithm 1: AdaBoost Ensemble Learning

---

**Input** : Initial dataset  $D_1$  with  $N$  examples, number of iterations  $k$

**Output**: Ensemble classifier  $H$

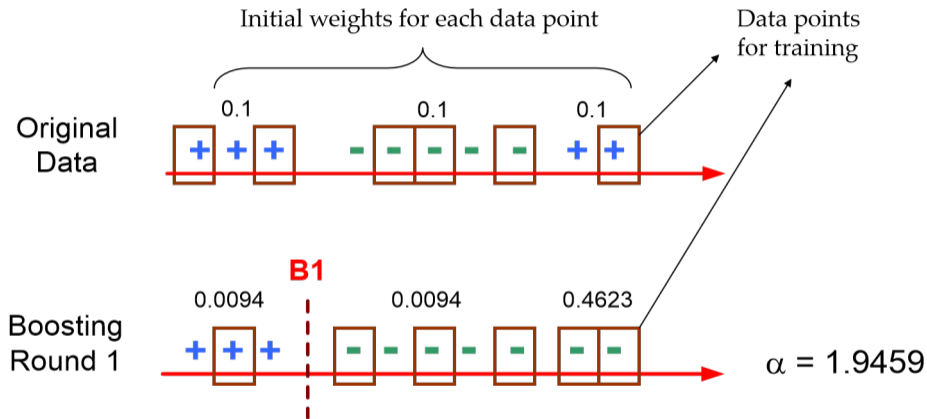
```

1 Initialize equal weights:  $w_j^{(1)} \leftarrow \frac{1}{N}$  for all  $j = 1$  to  $N$ 
2 for  $i \leftarrow 1$  to  $k$  do
3   Train classifier  $C_i$  on  $D_i$  using current weights  $\mathbf{w}^{(i)}$ 
4   Compute classifier weight:  $\alpha_i \leftarrow \frac{1}{2} \ln \left( \frac{1-\epsilon_i}{\epsilon_i} \right)$  //  $\epsilon_i$  is the weighted error of  $C_i$ 
5   Update weights for each example  $j$ :
6      $w_j^{(i+1)} \leftarrow w_j^{(i)} \cdot \exp(-\alpha_i \cdot y_j \cdot C_i(\mathbf{x}_j))$ 
7   Normalize weights:  $\mathbf{w}^{(i+1)} \leftarrow \mathbf{w}^{(i+1)} / \sum_j w_j^{(i+1)}$ 
8   Create  $D_{i+1}$  by weighted sampling from  $D_i$  with weights  $\mathbf{w}^{(i+1)}$ 
9 end
10 Construct ensemble classifier:  $H(\mathbf{x}) = \left( \sum_{i=1}^k \alpha_i C_i(\mathbf{x}) \right)$ 

```

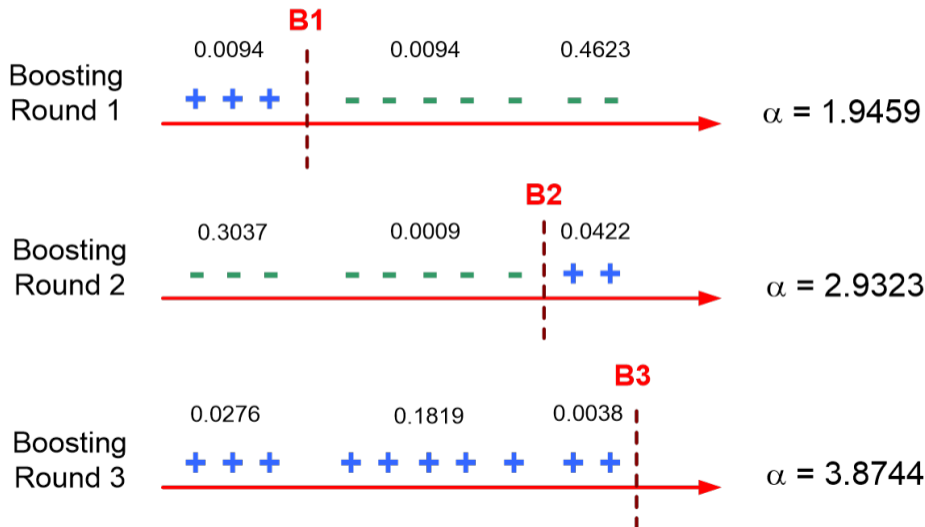
---

# Illustrating AdaBoost



Video Containing Introduction to AdaBoost:  
[http://videlectures.net/mlss05us\\_rudin\\_da/](http://videlectures.net/mlss05us_rudin_da/)

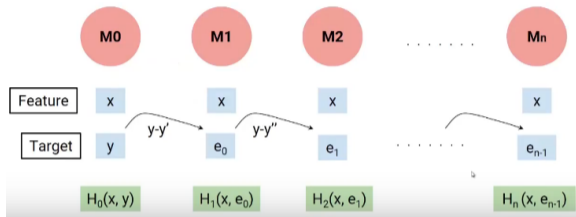
# Illustrating AdaBoost



# Gradient Boosting

- Often used with decision trees and regression
- Common winner in task competitions
- Train first model  $F_1$  with the basic training set
- Train next model  $h$  creating updated ensemble model  $F_{m+1} = F_m + h$
- But, train  $h$  using the residual/error (the difference between the target and the current output of  $F_m$ )
  - For  $h$ , change training instance  $(x, y)$  to  $(x, y - F_m(x))$
  - Each new model learns to output and cancel the remaining error from the previous model, leaving less error with each model
  - Learning focuses on instances where the latest  $F_m$  has higher error
- Also learns each models weighting coefficient  $\gamma$  with gradient descent to minimize chosen loss function (SSE common)
- Once trained,  $F_m$  no longer changes, and we keep adding new  $h$ s until remaining error is almost gone or test error begins to increase

# Gradient Boosting



The diagram illustrates the cumulative function  $F_n(x)$  in Gradient Boosting. It shows a sequence of models  $M_0, M_1, M_2, \dots, M_n$ . The cumulative function is built up step by step:

$$F_0(x) = H_0(x, y) + e_0$$

$$F_1(x) = F_0(x) + H_1(x, e_0) + e_1$$

$$F_2(x) = F_1(x) + H_2(x, e_1) + e_2$$

$$\vdots$$

$$F_n(x) = F_{n-1}(x) + H_n(x, e_{n-1}) + e_n$$

# Gradient Boosting

$$\begin{aligned}
 F_{n+1}(X) &= F_n(X) + \gamma_n H(x, e_n) \\
 F_0(X) &= \gamma_0 H_0(x, y) + e_0 \\
 F_1(X) &= F_0(X) + \gamma_1 H_1(x, e_0) + e_1 \\
 F_2(X) &= F_1(X) + \gamma_2 H_2(x, e_1) + e_2 \\
 &\vdots \\
 F_n(X) &= F_{n-1}(X) + \gamma_n H_n(x, e_{n-1}) + e_n
 \end{aligned}$$

- How to combine? Each models weighting/voting coefficient  $\gamma_i$  is learned with gradient descent to minimize loss as models are created, then frozen

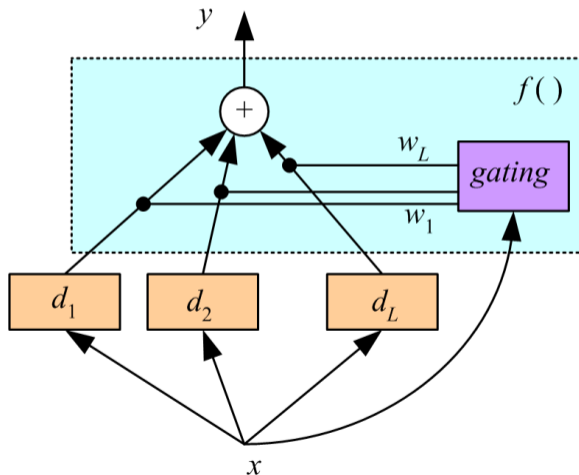


# An Example of Gradient Boosting

- Fit a shallow regression tree  $T_1$  to the data
  - the first model is  $M_1 = T_1$
  - The shortcomings of the model are given by the negative gradients.
- Fit a tree  $T_2$  to the negative gradients
  - The second model is:  $M_2 = M_1 + \eta \cdot \gamma_2 \cdot T_2$
  - $\eta$  is a learning rate to encourage more models
  - $\gamma_i$  is optimized, then frozen, so that  $M_i$  best fits the data
- Continue adding models (trees) until stopping criteria met
- The final model is  $M_{\text{final}} = M_{\text{final}-1} + \eta \cdot \gamma_{\text{final}} \cdot M_{\text{final}}$

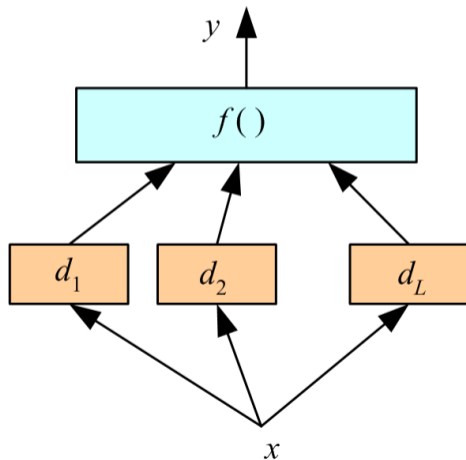
# Voting

Voting where weights are input-dependent (gating) [2]



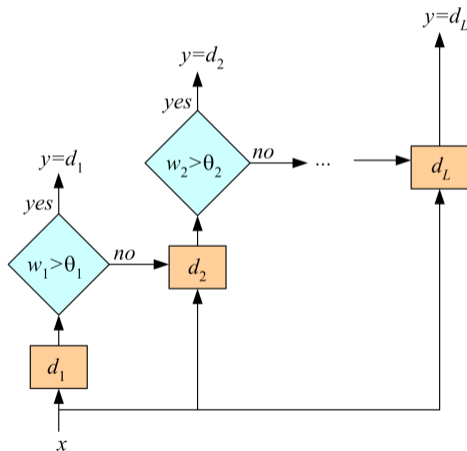
# Stacking

Combiner  $f()$  is another learner [3]



# Cascading

Use  $d_j$  only if the preceding ones are not confident  
Cascade learners in order of complexity



# Outline

- Introduction
- Types & Popular Algorithms
- **Challenges & Current Research Progress**
- Future Directions
- Summary

# What is the Main Challenge for Developing Ensemble Models?

- The main challenge is **not** to obtain **highly accurate base models**, but rather to **obtain base models that make different kinds of errors**.
  - For example, if ensembles are used for classification, high accuracies can be accomplished **if different base models misclassify different training examples**, even if the base classifier accuracy is low. Independence between two base classifiers can be assessed in this case by measuring the degree of overlap in training examples they misclassify ( $|A \cap B| / |A \cup B|$ )—more overlap means less independence between two models.
- Interpretability: Harder to interpret than single models.
- Model Selection: Choosing base learners and ensemble strategy.
- Computational Cost: Ensembles require more resources.

## Current Research Landscape

Ensemble learning has firmly established itself as a powerful strategy within machine learning, integrating multiple base learners such as bagging, boosting, and stacking to achieve superior predictive accuracy compared to single classifiers [4]. Classic algorithms, including Random Forests, AdaBoost, Gradient Boosting, XGBoost, LightGBM, and CatBoost, remain popular due to their effectiveness and relatively straightforward implementation [5].

Recently, ensemble deep learning, which leverages the power of neural networks through methods like deep model averaging and fusion, has become increasingly significant.

Techniques such as weight averaging, mode connectivity, and ensemble outputs have been explored to improve prediction performance and model generalization, although they incur considerable computational and memory costs [6, 7].

Novel ensemble variants such as Extremal Ensemble Learning (EEL) have emerged, particularly useful in graph-related tasks like clustering and community detection [8]. Concurrently, reinforcement learning ensembles have seen growing interest, employing multiple agents or policies to stabilize and enhance performance in complex decision-making scenarios [9].

# Outline

- Introduction
- Types & Popular Algorithms
- Challenges & Current Research Progress
- **Future Directions**
- Summary



# Future Directions

Ensemble learning continues to evolve with several promising research directions. An important area is the extension of ensemble methods to large language models (LLMs) and foundation models, necessitating techniques that handle vast parameter spaces efficiently [7]. Furthermore, the trade-off between efficiency and performance will remain a crucial area, spurring advancements in lightweight ensembles, model pruning, and efficient fusion methods [10].

Federated and distributed ensemble learning represents another burgeoning area of research. In privacy-sensitive contexts, ensembles must aggregate knowledge from decentralized datasets without compromising data privacy, emphasizing communication efficiency and robustness [10]. Finally, increased attention on explainability, fairness, and responsible AI will drive research toward developing transparent, interpretable, and ethically responsible ensemble methods, particularly in regulated fields like finance, healthcare, and autonomous systems [10].






# Outline

- Introduction
- Types & Popular Algorithms
- Challenges & Current Research Progress
- Future Directions
- **Summary**





# Summary

- Ensemble approaches use multiple models in their decision making. They frequently accomplish high accuracies, are less likely to over-fit and exhibit a low variance. They have been successfully used in the Netflix contest and for other tasks. However, some research suggest that they are sensitive to noise ([http://www.phillong.info/publications/LS10\\_potential.pdf](http://www.phillong.info/publications/LS10_potential.pdf)).
- The key of designing ensembles is diversity and not necessarily high accuracy of the base classifiers: Members of the ensemble should vary in the examples they misclassify. Therefore, most ensemble approaches, such as AdaBoost, seek to promote diversity among the models they combine.
- The trained ensemble represents a single hypothesis. This hypothesis, however, is not necessarily contained within the hypothesis space of the models from which it is built. Thus, ensembles can be shown to have more flexibility in the functions they can represent. Example: [http://www.scholarpedia.org/article/Ensemble\\_learning](http://www.scholarpedia.org/article/Ensemble_learning)
- Current research on ensembles centers on: more complex ways to combine models, understanding the convergence behavior of ensemble learning algorithms, parameter learning, understanding over-fitting in ensemble learning, characterization of ensemble models, sensitivity to noise.

## Reference I

-  L. Breiman, “Bagging predictors,” *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.
-  R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, “Adaptive mixtures of local experts,” *Neural computation*, vol. 3, no. 1, pp. 79–87, 1991.
-  D. H. Wolpert, “Stacked generalization,” *Neural networks*, vol. 5, no. 2, pp. 241–259, 1992.
-  T. G. Dietterich, “Ensemble methods in machine learning,” in *Multiple Classifier Systems*, ser. Lecture Notes in Computer Science, J. Kittler and F. Roli, Eds. Berlin, Heidelberg: Springer, 2000, vol. 1857, pp. 1–15.
-  G. Seni and J. F. Elder, *Ensemble methods in data mining: improving accuracy through combining predictions*, ser. Synthesis Lectures on Data Mining and Knowledge Discovery. Morgan & Claypool Publishers, 2010, vol. 2, no. 1.

## Reference II

-  G. Huang, Y. Li, G. Pleiss, Z. Liu, J. E. Hopcroft, and K. Q. Weinberger, “Snapshot ensembles: Train 1, get m for free,” in *International Conference on Learning Representations (ICLR)*, Toulon, France, Apr. 2017. [Online]. Available: <https://openreview.net/forum?id=BJYwwY9II>
-  Y. Fu, S. Xu, and Y. Wang, “Deep model fusion: A survey,” *arXiv preprint arXiv:2309.15698*, 2023. [Online]. Available: <https://arxiv.org/abs/2309.15698>
-  C. T. Kelley, “Extremal ensemble learning for graph partitioning,” *SIAM Journal on Scientific Computing*, vol. 42, no. 3, pp. A1596–A1618, 2020.
-  Y. Zhang, Q. Huang, Z. Chen, X. Zhang, and H. Yao, “Ensemble reinforcement learning: A survey,” *arXiv preprint arXiv:2303.02618*, 2023. [Online]. Available: <https://arxiv.org/abs/2303.02618>

## Reference III



B. Cao, C. Zhang, and P. S. Yu, “A comprehensive survey on ensemble multi-featured deep learning models: Applications, challenges, and future directions,” *Information Fusion*, vol. 89, pp. 1–27, 2023.