Ensemble Learning (二)

Substract:

- Voting, Bagging and Pasting --> Decision Tree, Random Forest
- Boosting(Adaboost, GBDT), Stacking --> Xgboost, LightGBM

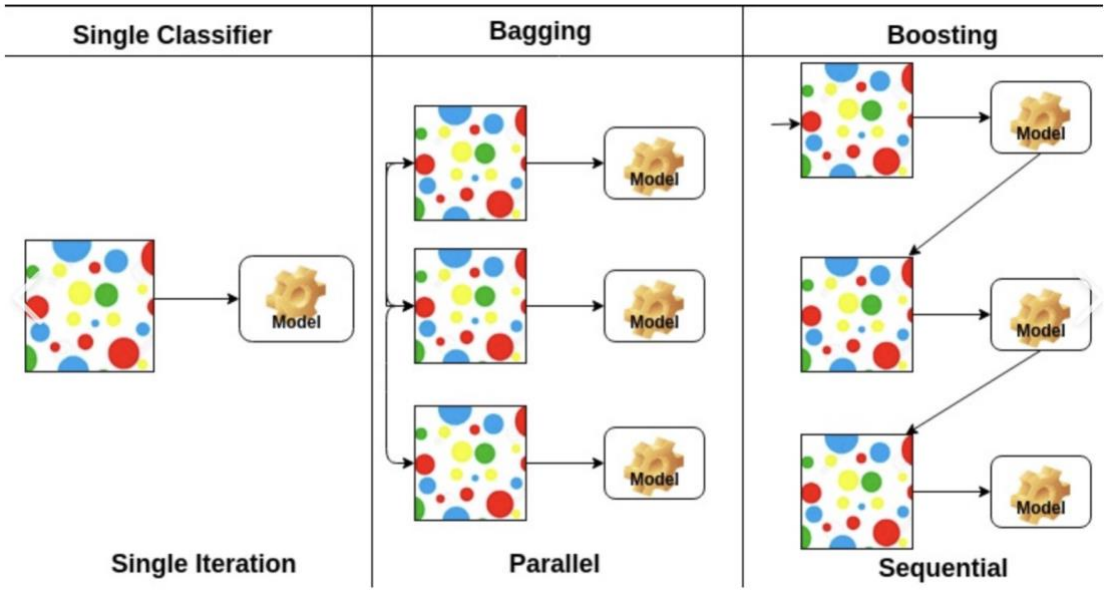
Wisdom of the crowd (aggregated > individual)

Ensemble: a group of predictors (weak estimators --> strong estimators)

Insights: Ensemble methods -- an aggregated predictor

Boosting:

- Train predictors sequentially, each trying to correct its predecessor
- Can't be trained parallelly

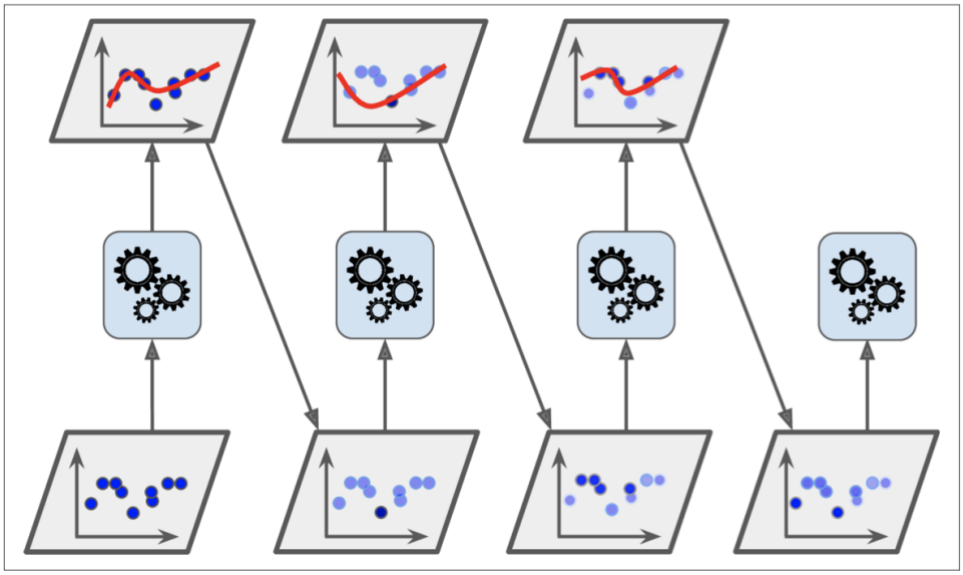


一、 AdaBoost（Adaptive Boosting）

Target： training instances that the predecessor underfitted

Training Steps:

- A first base classifier is trained and used to make predictions on the training set
- The relative weight of misclassified training instances is then incresed
- A second classifier is trained using the updated weights on the training set and …



Maths:

- **E1: Weighted error rate of the j[th] predictor (r_j)**

$$r_j = \frac{\sum_{\substack{i=1 \\ \hat{y}_j^{(i)} \neq y^{(i)}}}^{m} w^{(i)}}{\sum_{i=1}^{m} w^{(i)}}$$

where $\hat{y}_j^{(i)}$ is the j[th] predictor's prediction for the i[th] instance; $w^{(i)}$ is each instance weight, initially set to $\frac{1}{m}$; m is the number of instances.

- **E2: Predictor weight (α_j)**

$$\alpha_j = \eta \log \frac{1 - r_j}{r_j}$$

where η is the learning rate hyperparameter , defaults to 1.

- **E3: Weight update rule ($w^{(i)}$)**

$$\text{for } i = 1, 2, \cdots, m$$

$$w^{(i)} \leftarrow \begin{cases} w^{(i)} & \text{if } \hat{y}_j^{(i)} = y^{(i)} \\ w^{(i)} \exp\left(\alpha_j\right) & \text{if } \hat{y}_j^{(i)} \neq y^{(i)} \end{cases}$$

The instance weights are updated, the misclassified instances are boosted.

- **E4: Normalization of the instance weight**

$$\frac{w^{(i)}}{\sum_{i=1}^{m} w^{(i)}}$$

The new predictor is trained using the updated weights and the whole process is repeated.

- **E5: AdaBoost predictions**

$$\hat{y}(\mathbf{x}) = \underset{k}{\operatorname{argmax}} \sum_{\substack{j=1 \\ \hat{y}_j(\mathbf{x}) = k}}^{N} \alpha_j$$

  where *N* is the number of predictors.
  AdaBoost computes the predictions of all the predictors and weighs them using the predictor weight α_j . The predicted class is the one that receives the majority of weighted votes.

Notice:

- Compared with Gradient Descent, AdaBoost adds predictors to the ensemble to make the result model better instead of tweaking a single predictor's parameters to minimize a cost function.
- Once all predictors are trained, the ensemble makes predictions very much like bagging or pasting, except that predictors have different weights depending on their overall accuracy on the weighted training set.
- SAMME (Stagewise Additive Modeling using a Multiclass Exponential loss function) is the multiclass version of AdaBoost.
- SAMME.R(R stands for "Real") relies on class probabilities rather than predictions

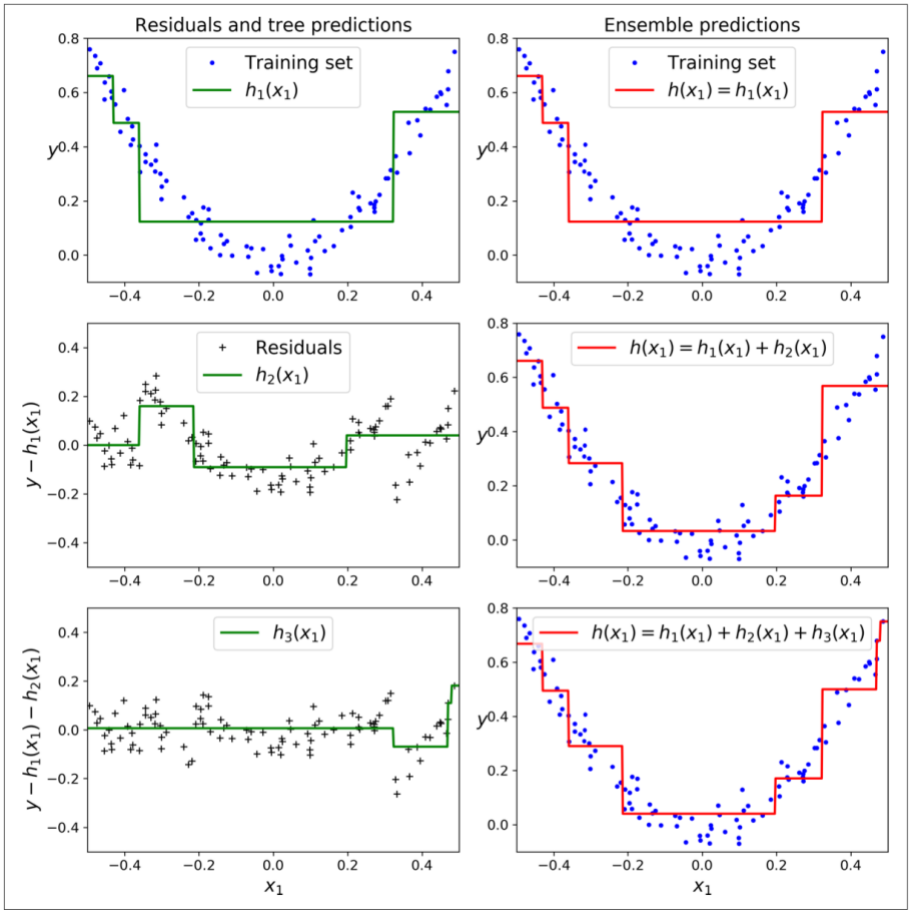
二、 Gradient Boosting

Target： redisudual errors made by the previous predictor

GBRT (Gradient Boosted Regression Trees): using Decision Trees as the base predictors

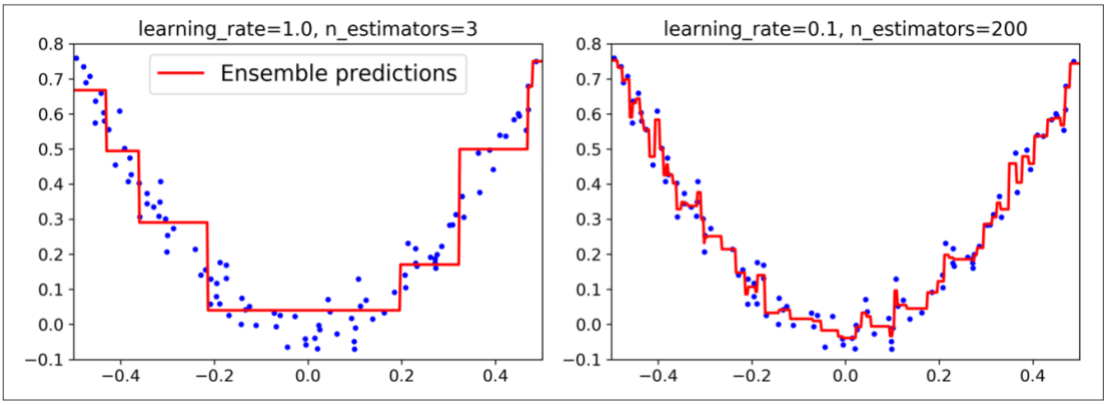
Training steps:

- Fit the new predictor to the residual errors made by the previous predictor

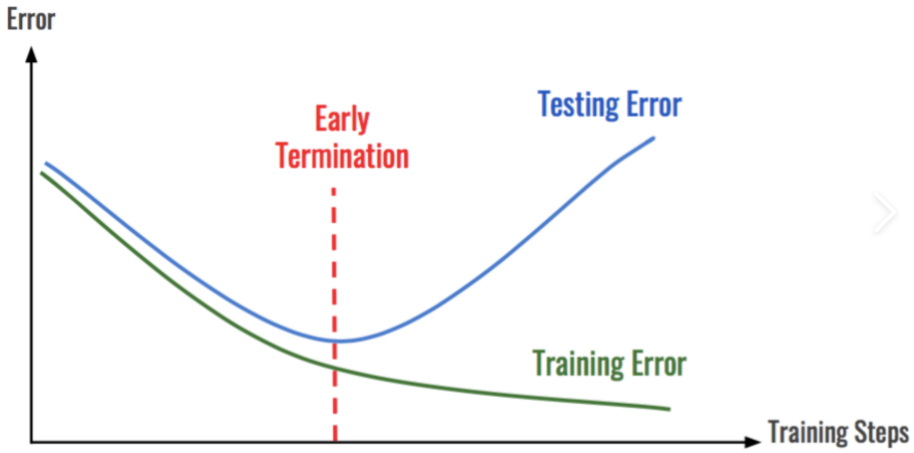


Hyperparameter:

- Learning_rate : scales the contribution of each tree
  - Regularization technique: shrinkage
    - Tradeoff: low learning_rate --> more trees --> generalize better
- Subsample : the fraction of training instances to be used for training each tree
  - Trades a higher bias for a lower variance
  - Speeds up training



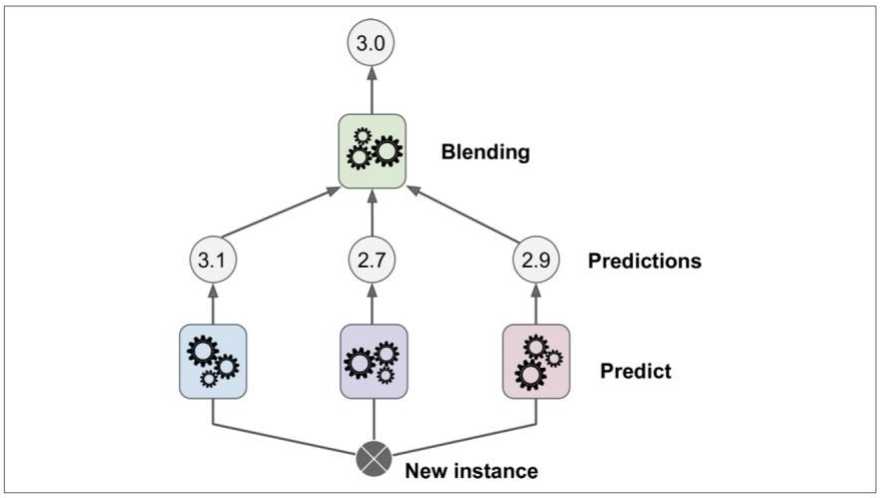
Early stopping: prevent the model from overfitting, get the best model --> optimal number of trees



XGBoost: Extreme Gradient Boosting

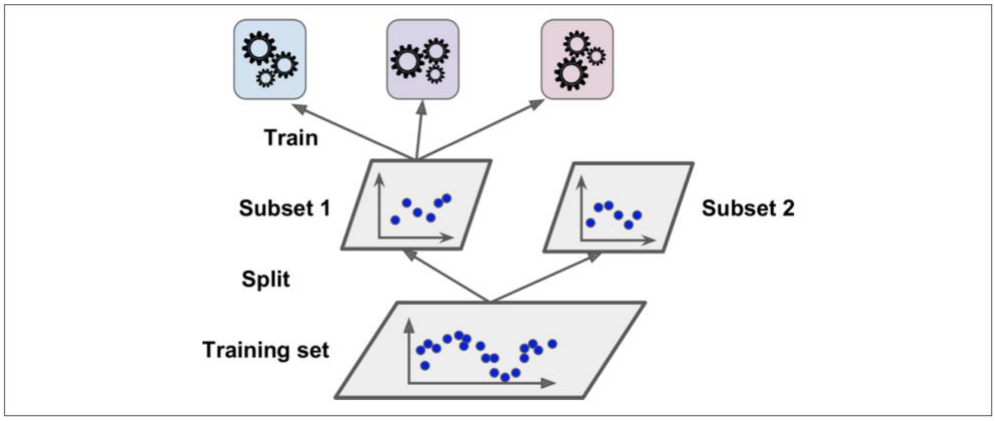
三、 Stacking (stacked generalization)

Idea: train a model to perform the aggreation rather than using trivial functions (such as hard voting) to aggregate the predictions

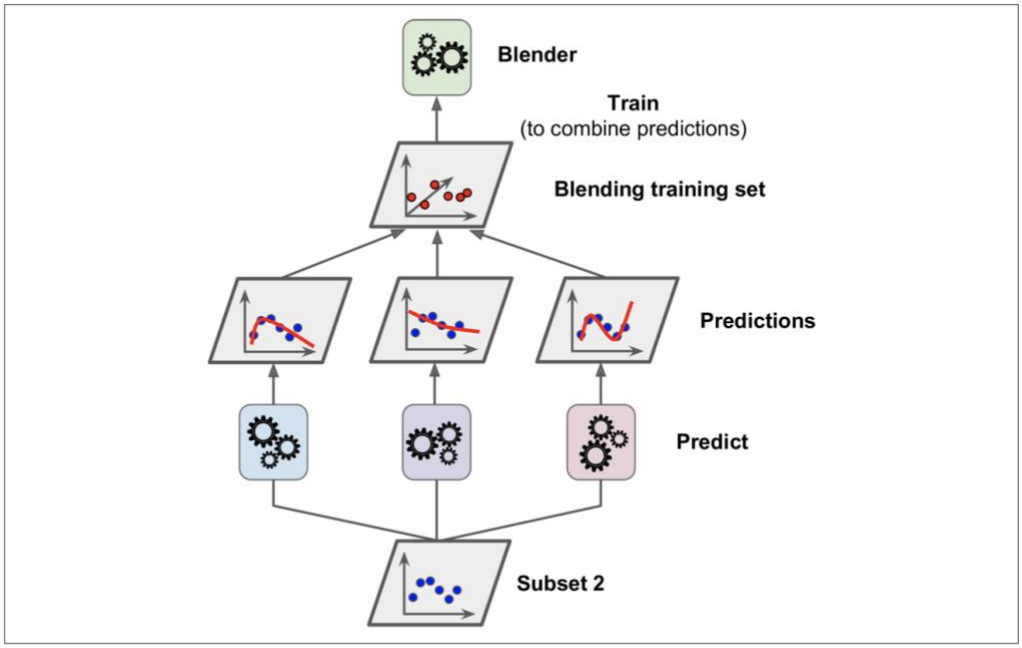


Training Process:

- Split the training set into two subsets
- Use the first subset to train the predictors in the first layer
- The first layer predictors are used to make predictions on the second subset (hold-out set) --> three predicted values for each instance --> new training set three-dimensional (3 features + 1 target value)
- Train the blender on this new training set



Train the first layer



Train the blender