We find that the \( t \)-layer casForest model is defined by a recursive formula:

\[
f(x) = \begin{cases} 
\phi_1(x) = \phi_1([x_{f_1}(x)]) & t = 1, \\
\phi_1(x) + f_s(x) & t > 1, \\
\phi_1(x) + f_{f_{t-1}}(x) & t > 1.
\end{cases}
\]

The entire additive cascade model is defined as follows:

\[
h(x) = \phi_1([x_{f_1}(x)]) = \phi_1 \left( \sum_{t=1}^{T} \alpha_t \phi_t(x) \right).
\]

The casForest model can be formalized as follows. We use a quadruple form:

- Forest block: \( \phi = (\phi_0, \phi_1, \ldots, \phi_T) \)
- casForest: \( h = [h_1, h_2, \ldots, h_T] \)
- Augmented feature: \( f = (f_1, f_2, \ldots, f_T) \)
- Sample distribution: \( D = (D_1, D_2, \ldots, D_T) \)

\( \phi_t \) is the function returned by the random forests block (Algorithm 1).

**Algorithm 1 Random forests block \( \mathcal{A}_D \) [33]**

Input: A training set \( S \) drawn from \( D \) and the augmented feature \( f_{x_{f_1}(x)}, \forall i \in m \).

Output: The function computed by the random forests block in the \( t \)-th layer: \( \phi_t \).

1. Divide \( S \) to \( b \)-fold subsets \( (S_1, S_2, \ldots, S_b) \) randomly.
2. for \( S_t \) in \( (S_1, S_2, \ldots, S_b) \)
3. using \( S_t \) to train two random forests and two completely random forests.
4. Compute the prediction rate \( \hat{p}(j) \) for \( f_{x_{f_1}(x)} \) of the first layer node generated by \( S_t / S_t \).
5. \( \alpha_t = \hat{p}(j) \) for any training sample \( (x, y) \). \( S_t \)
6. end for
7. \( \phi_t(x_{f_1}(x)) = \beta_t \hat{p}(j) \) for any test sample \( (x, y) \).
8. return \( \phi_t \)

**Background**

By realizing that the essence of deep learning lies in the layer-by-layer processing, in-model feature transformation, and sufficient model complexity, recently Zhou & Feng propose the deep forest model and the gcForest algorithm to achieve forest representation learning. It can achieve excellent performance on a broad range of tasks, and can even perform well on small or middle-scale of data.

**Cascade Forest**

The casForest model can be formalized as follows. We use a quadruple form:

- Forest block: \( \phi = (\phi_0, \phi_1, \ldots, \phi_T) \)
- casForest: \( h = [h_1, h_2, \ldots, h_T] \)
- Augmented feature: \( f = (f_1, f_2, \ldots, f_T) \)
- Sample distribution: \( D = (D_1, D_2, \ldots, D_T) \)

\( \phi_t \) is the function returned by the random forests block (Algorithm 1).

**Theorem 1.** Let \( D \) be a distribution over \( X \times Y \) and \( S \) be a training set of \( n \) samples drawn from \( D \). With probability at least \( 1 - \delta \), for \( r > 0 \), the strongly classifier \( F(x) \) (the \( t \)-layer casForest model) satisfies that

\[
\Pr[|F(x) - y| < \epsilon] \geq 1 - \exp \left( -\frac{2\epsilon^2}{2r^2} \right)
\]

where

\[
d = \frac{2}{1 - E[F(x)] + r/9}, \quad \epsilon = \ln n + \ln \ln n + \frac{t}{2} + \frac{2\epsilon^2}{2r^2} + \ln 2 + \lambda n^2.
\]

**Remark 1.** From Theorem 1, we know that the gap between the generalization error and empirical loss is generally bounded by the rate \( \frac{1}{\sqrt{n}} \sqrt{\ln n/\ln m} + \ln m/\ln n \), which shows minimizing the margin ratio is the key to good generalization.

**Remark 2.** The hypothesis term \( \sum_{t=1}^{T} \alpha_t \phi_t \) admits an explicit dependency on the mixture coefficients. Though some hypothesis sets used for learning could have large complexity, it will not be detrimental to generalization when the corresponding mixture weight is relatively small.

**Optimization**

Since we formulate casForest as an additive model, we utilize the reweighting approach to minimize the expected margin distribution loss.