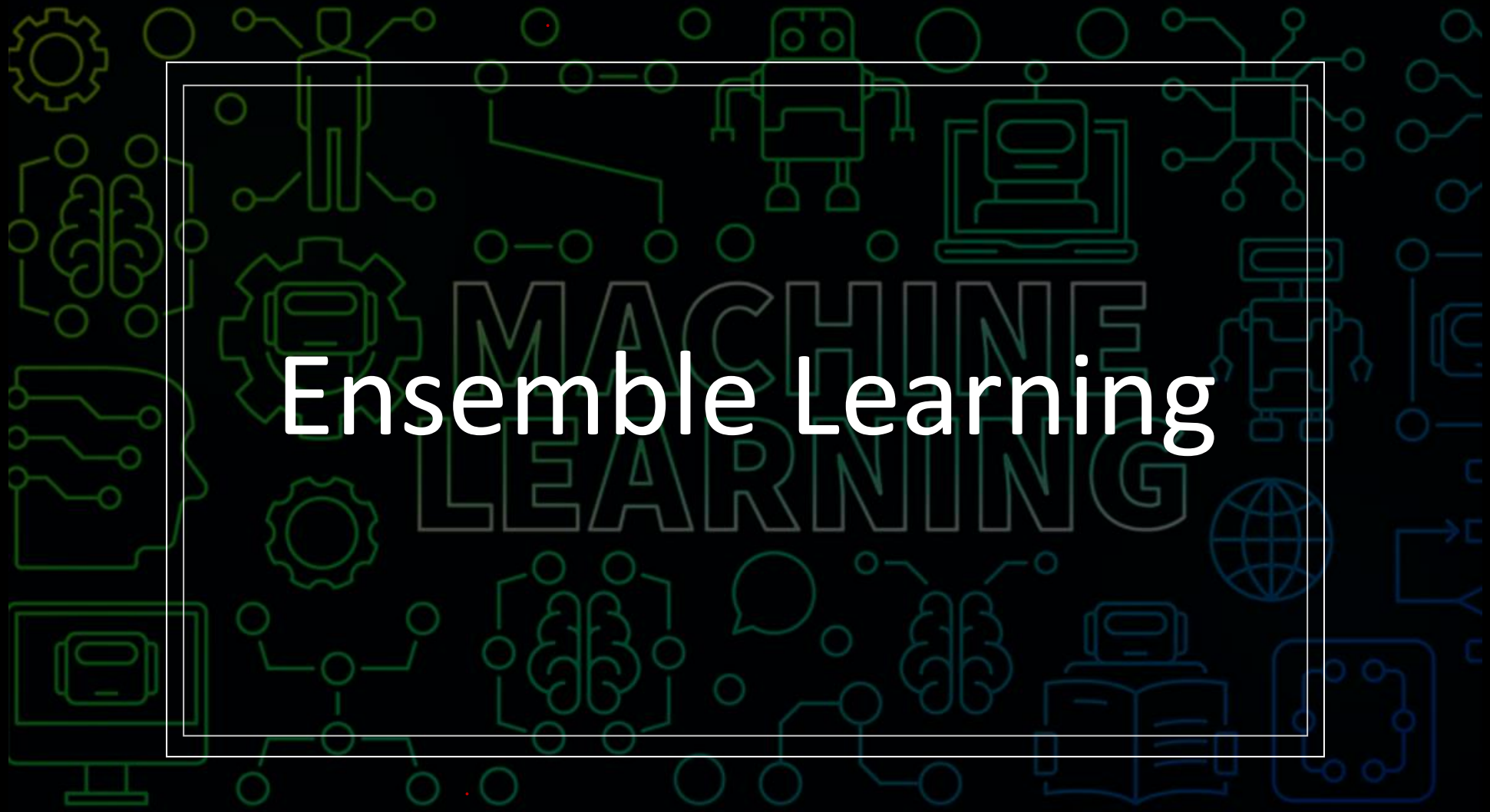
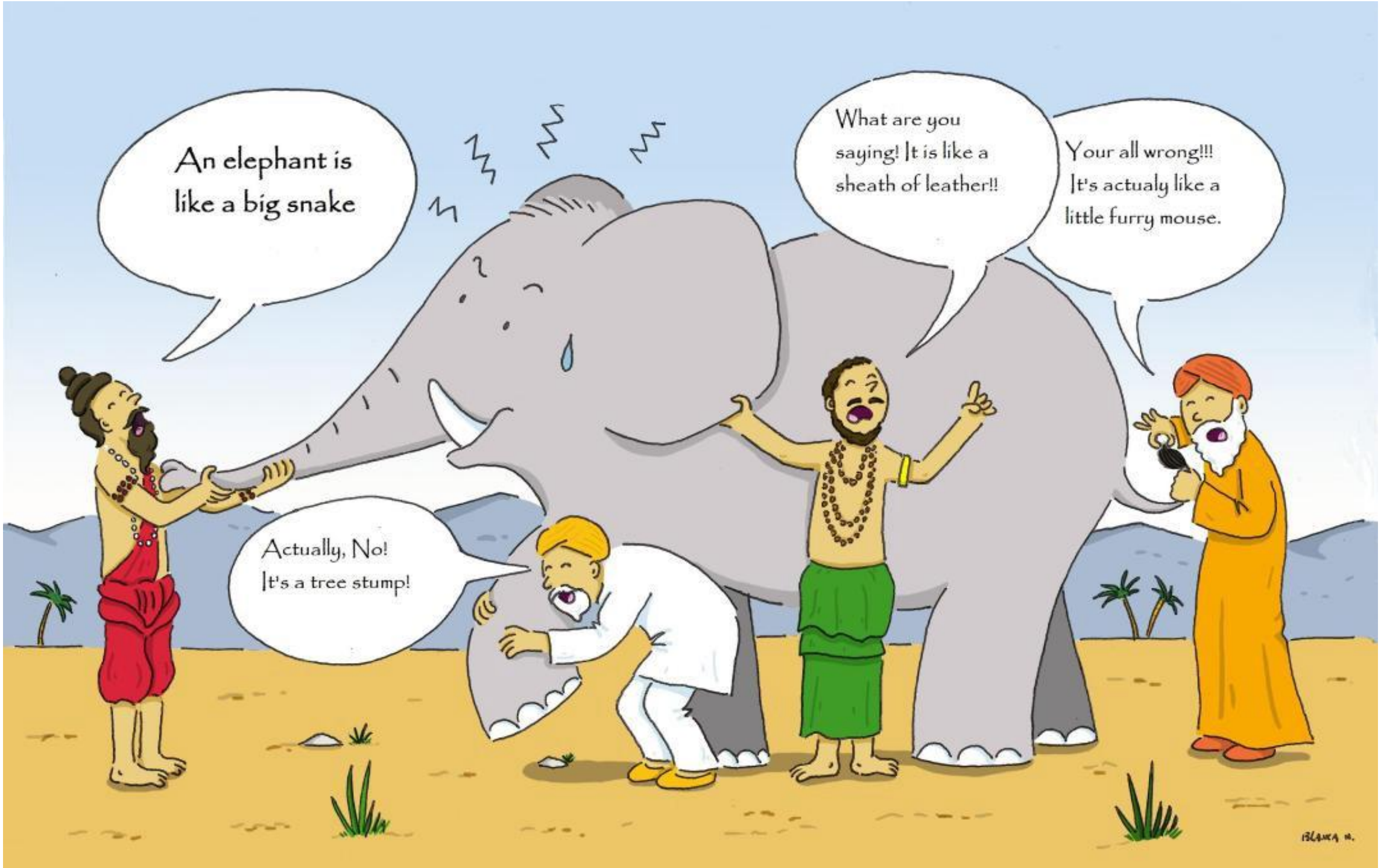


Ensemble Learning


An elephant is like a big snake
Actually, No! It's a tree stump!
What are you saying! It is like a sheath of leather!!
Your all wrong!!! It's actualy like a little furry mouse.

Ensemble Learning

Consider a set of classifiers h_1, ..., $\mathrm{h_L}$

Idea: construct a classifier $H(\mathbf{x})$ that combines the individual decisions of h_1, ..., $\mathrm{h_L}$

Ensemble Learning

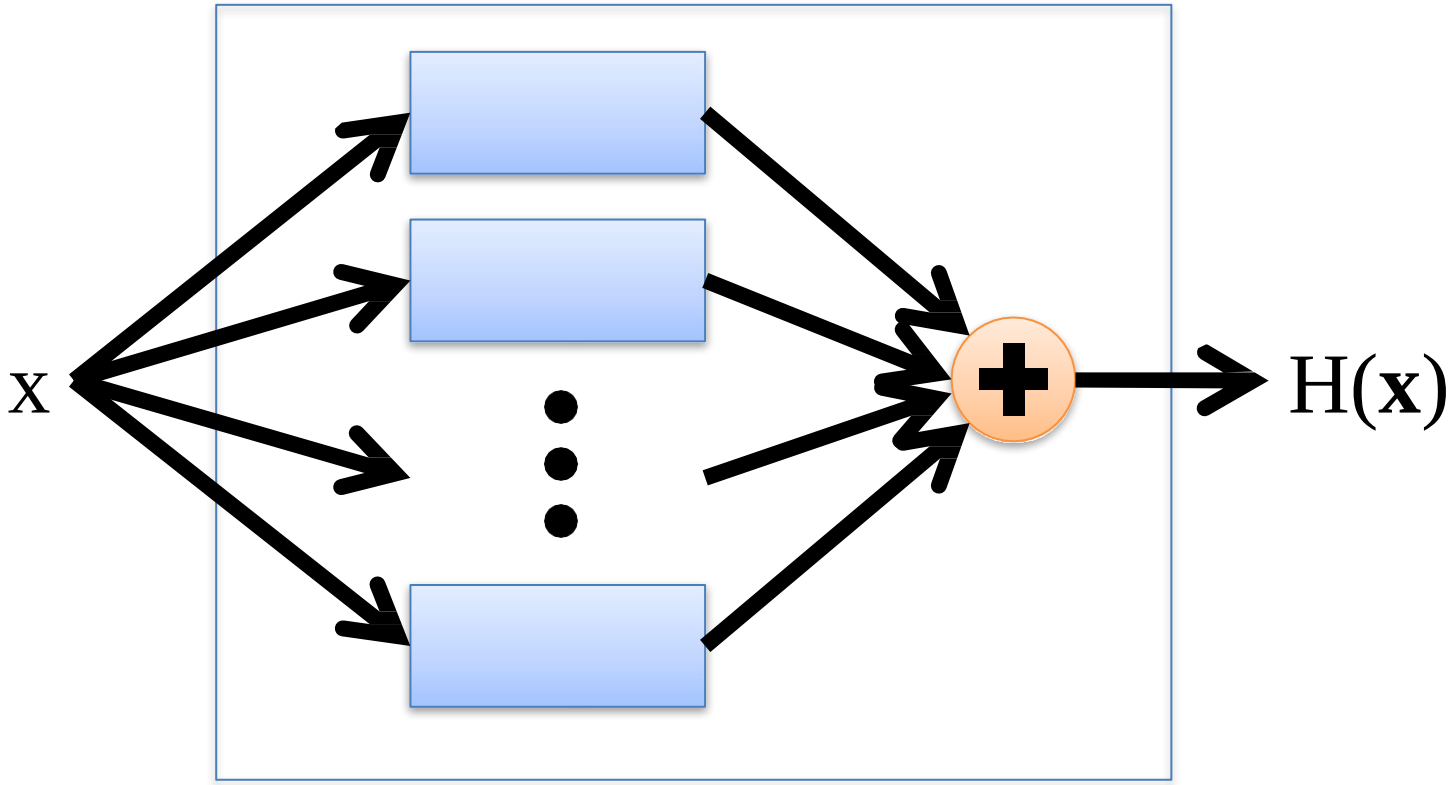
Consider a set of classifiers h_1, ..., $\mathrm{h_L}$

Idea: construct a classifier $H(\mathbf{x})$ that combines the individual decisions of h_1, ..., $\mathrm{h_L}$

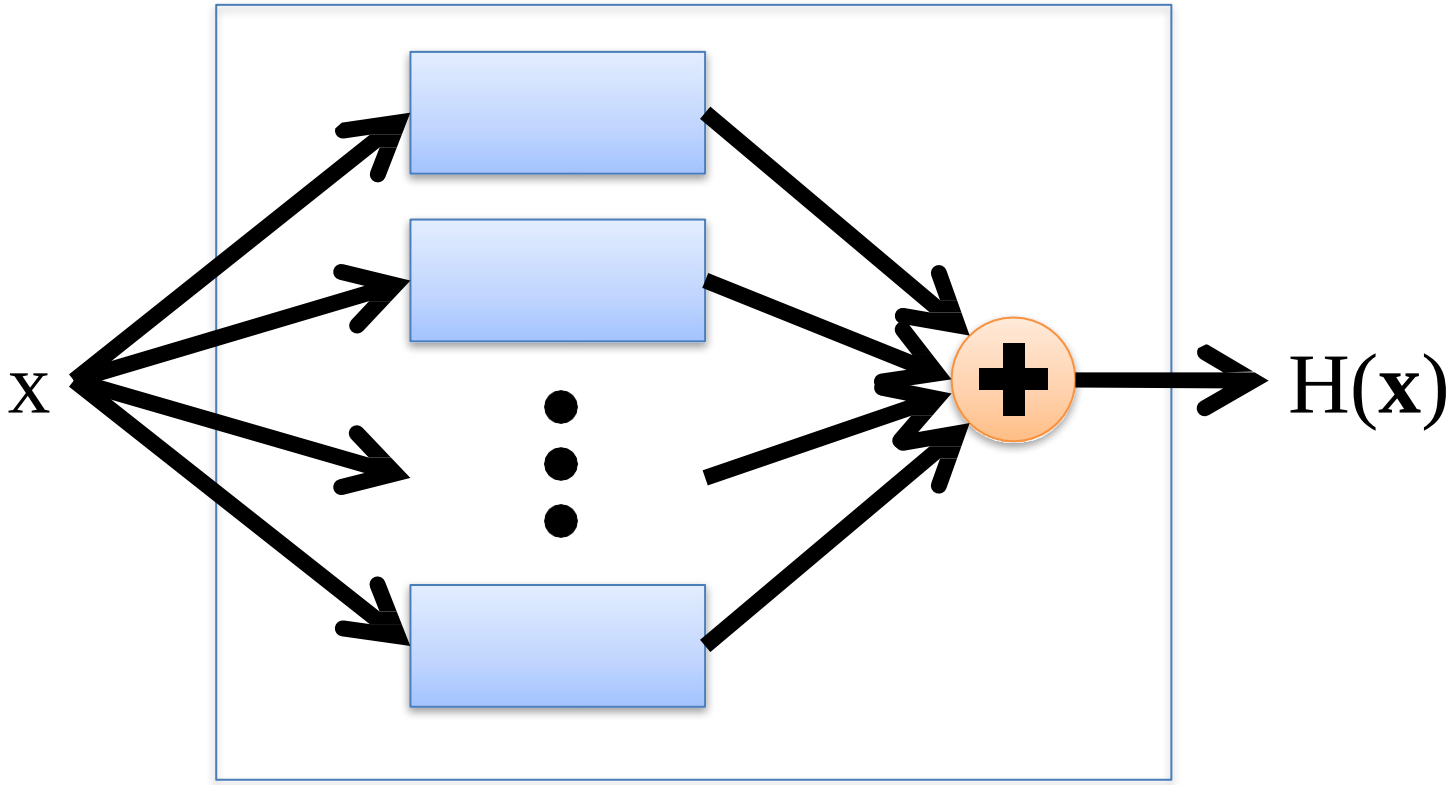
Successful ensembles require **diversity**

- Classifiers should make different mistakes
- Can have different types of base learners

Combining Regressors:Averaging

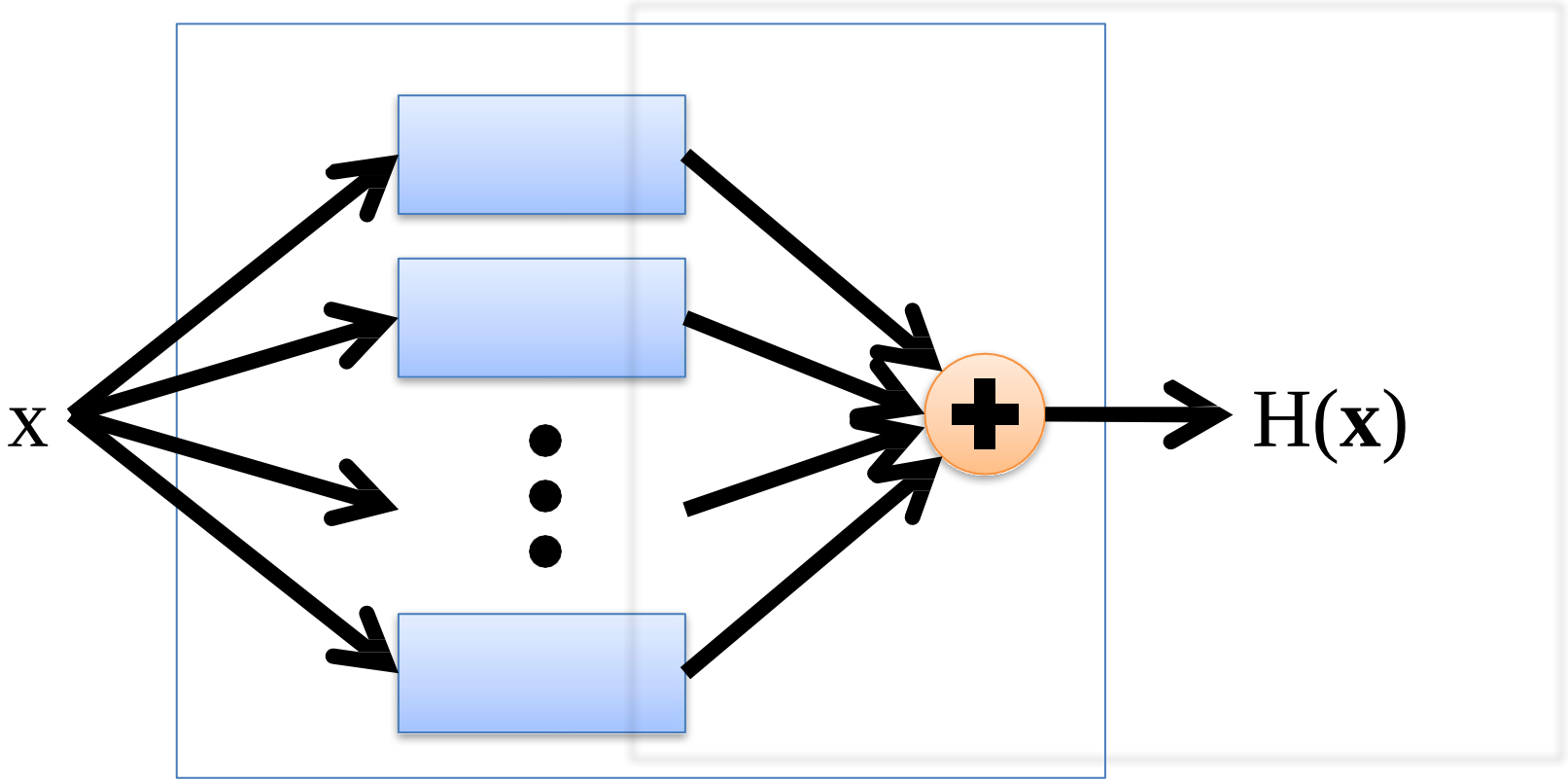


Combining Classifiers: Voting



- Final hypothesis is a simple vote of the members

Combining Classifiers: Voting



- Final hypothesis is a simple vote of the members

Hard vs Soft voting

Combining Classifiers Voting

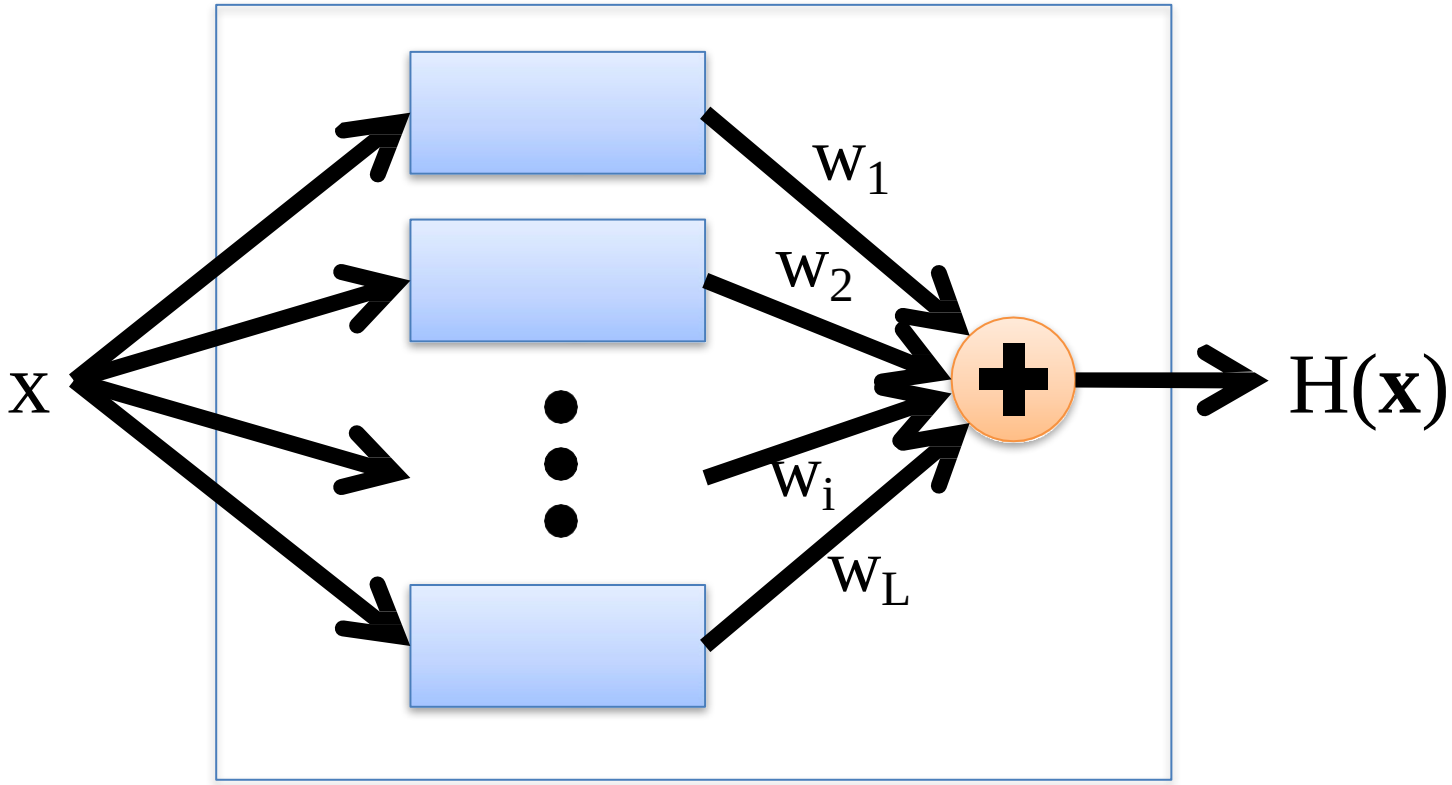
sklearn.ensemble.VotingClassifier(estimators, *, voting='hard', weights=None, n_jobs=None, flatten_transform=True, verbose=False)

voting*{'hard', 'soft'}, default='hard'*

If 'hard', uses predicted class labels for majority rule voting. Else if 'soft', predicts the class label based on the argmax of the sums of the predicted probabilities, which is recommended for an ensemble of well-calibrated classifiers.

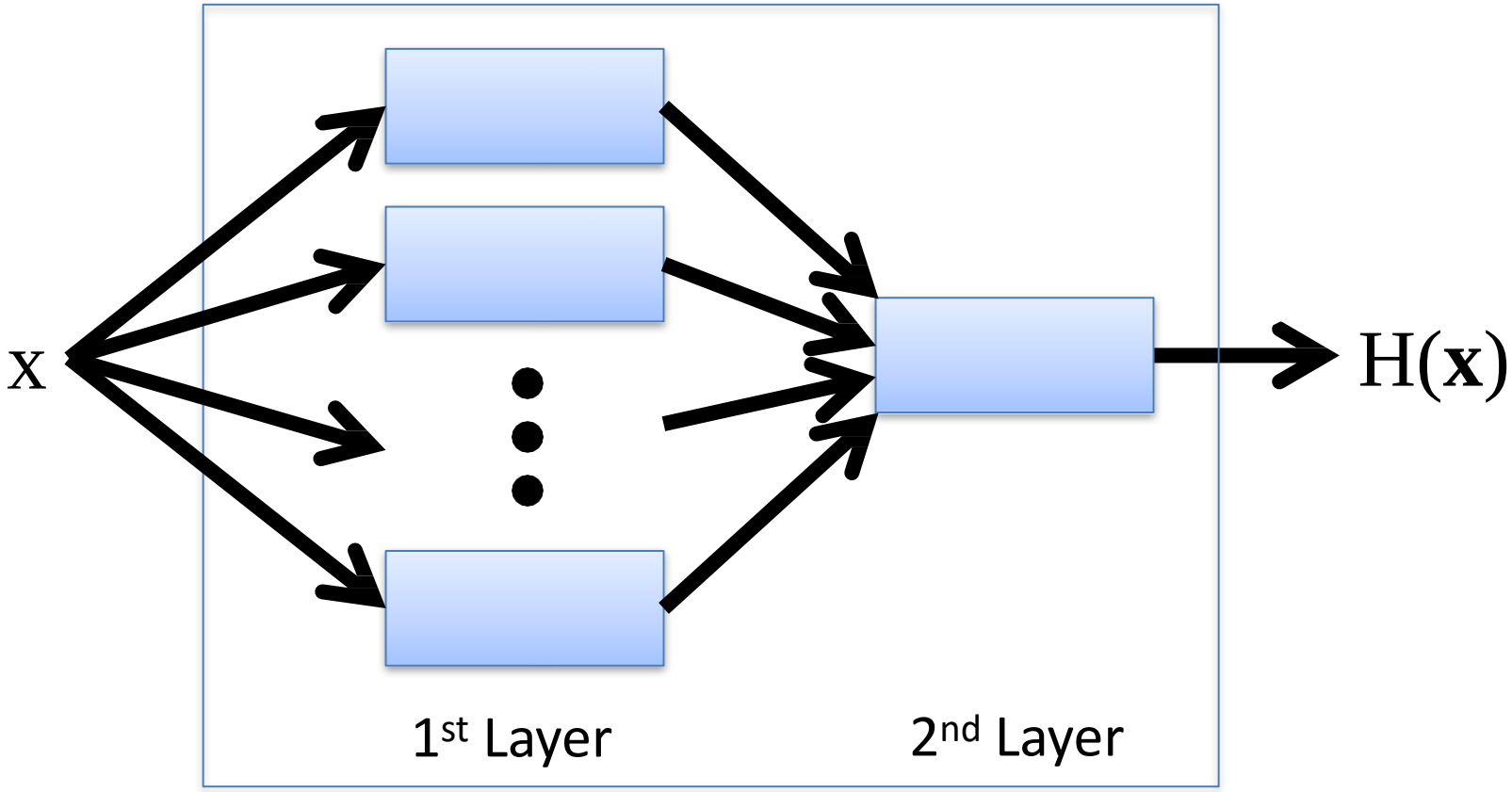
sklearn.ensemble.VotingClassifier(estimators, *, voting='hard', weights=None, n_jobs=None, flatten_transform=True, verbose=False)

Combining Classifiers: Weighted Average



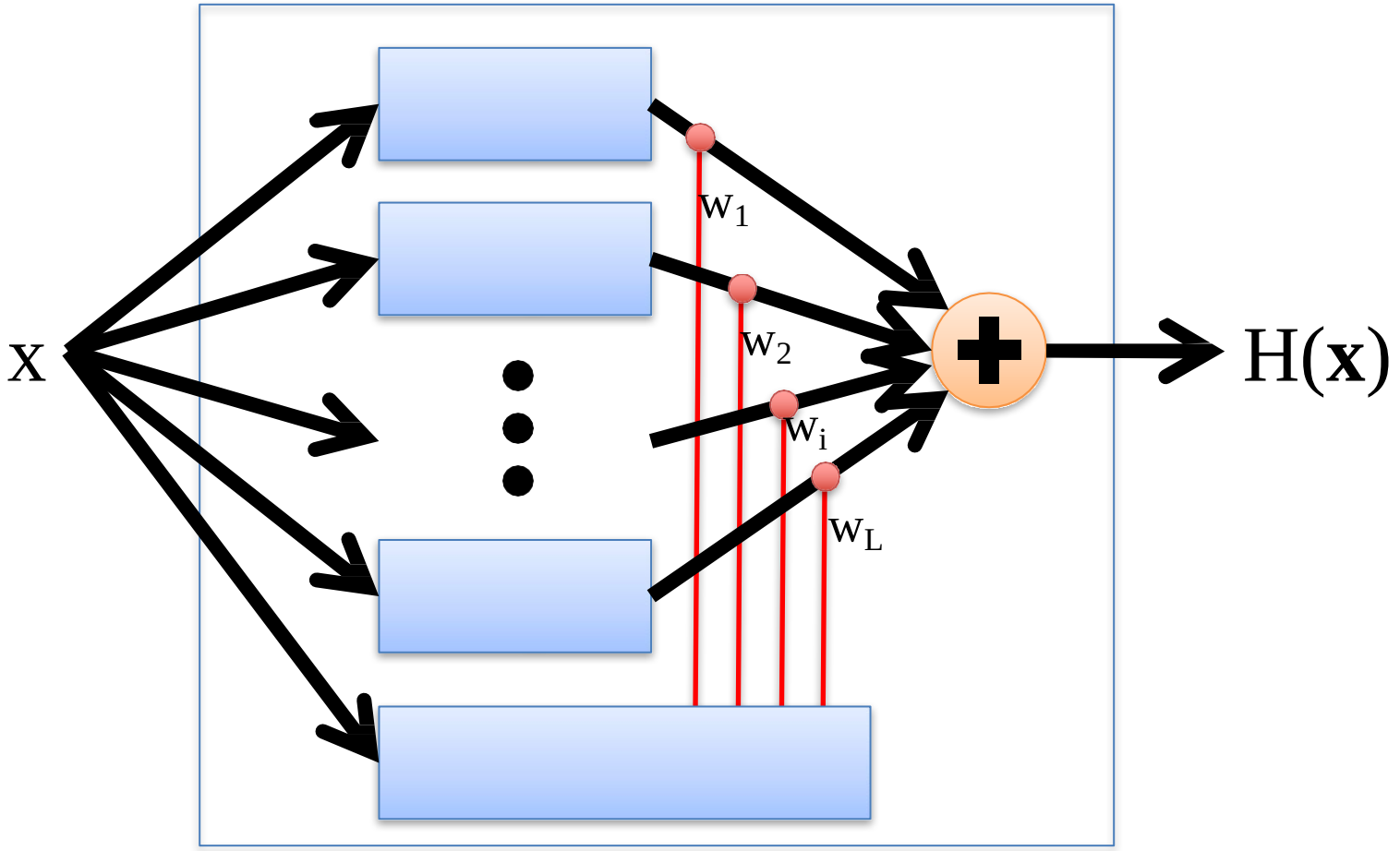
- Coefficients of individual members are trained using a validation set

Combining Classifiers: Stacking



- Predictions of 1st layer used as input to 2nd layer
- Train 2nd layer on validation set

Combining Classifiers: Gating



- Coefficients of individual members depend on input
- Train gating function via validation set

How to Achieve Diversity

Cause of the Mistake	Diversification Strategy
Pattern was difficult	Hopeless
Overfitting	Vary the training sets
Some features are noisy	Vary the set of input features

Manipulating the Training Data

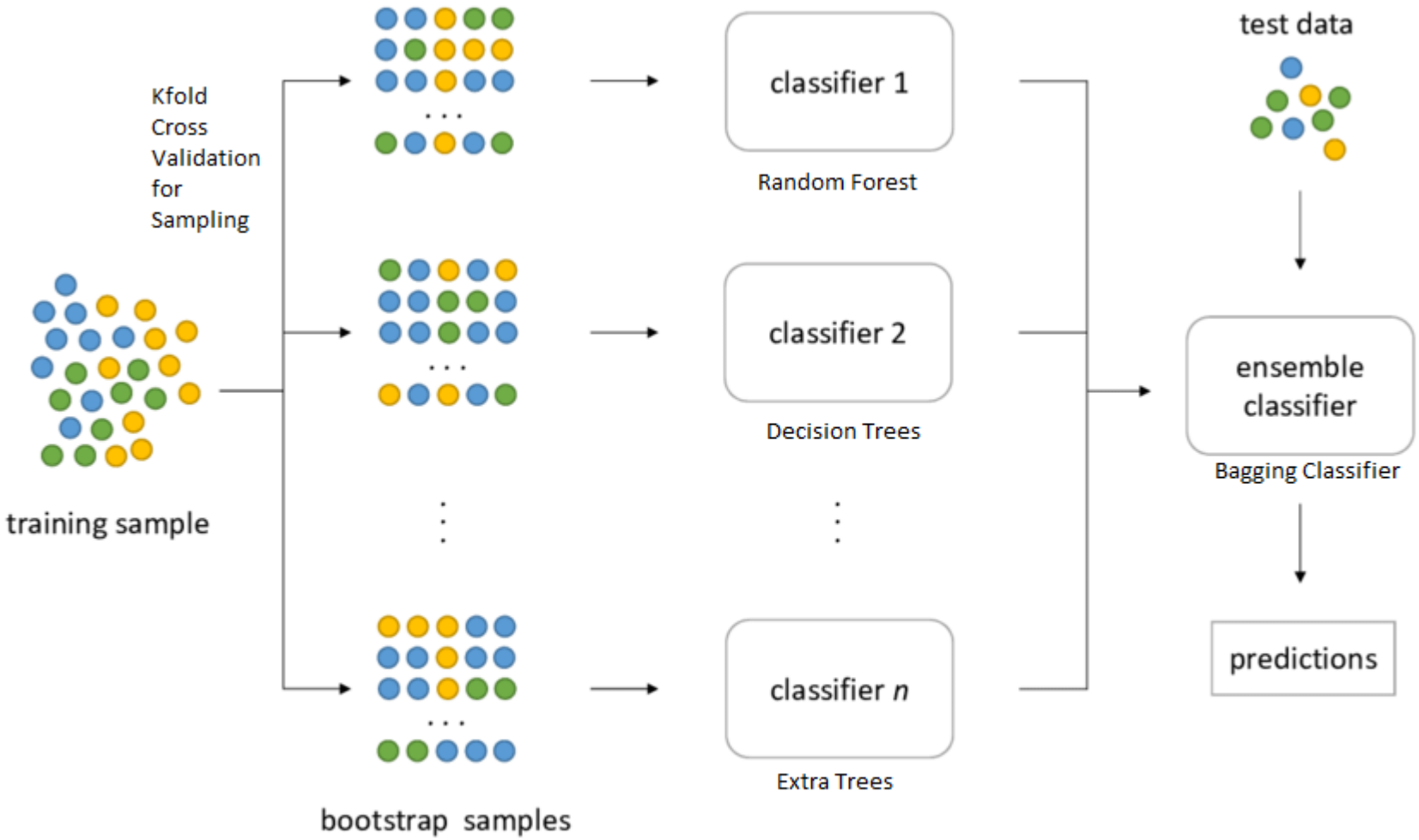
Bootstrap replication:

- Given n training examples, construct a new training set by sampling n instances with replacement
- Excludes ~30% of the training instances

Bagging:

- Create bootstrap replicates of training set
- Train a classifier (e.g., a decision tree) for each replicate
- Estimate classifier performance using out-of-bootstrap data
- Average output of all classifiers

Bagging Classifier



Bagging Classifier Process Flow

நன்றி வணக்கம்