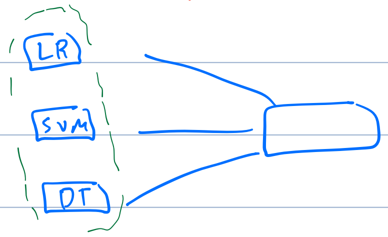
train different classifiers then combine them

weak learners
High bais, low variance

give better than individuals (not always) ①   robust to overfitting ②



ways to combine:-

① majority voting (Regression = average)

② wieghted majority voting classifier

③ add another classifier (second layer) train it using validation data → stacking

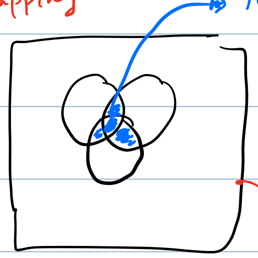
cratirea:- (to achive the goal)

① minimum acceptable error (better than Random) → week learner

② should these classifiers make error in different samples

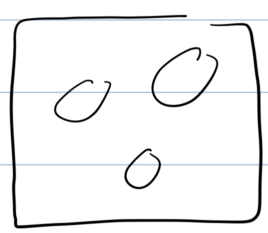
we want to min. overlapping

the ensemble error



usually occure

error for ensemble = 0%



hard to get that

hetro = SVM, DT...
hemo = Same classifier with different hyperparameter or with playing with data

to get min. overlapping:-

- manipulating the given data to every classifier (different from each other)
- use different features for every classifier
- use different classifiers
- use same classifiers with different hyper-parameters

different features is better than different classifiers

to reduce variance (overfitting)

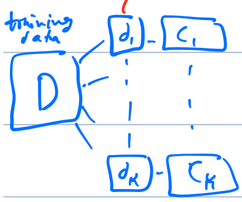
with homo classifiers

Bagging

→ better than just partetion the data

→ very good for decesion tree (since it's problem is easy overfit)

split data



D size = d_i size

take from D then remove and duplicate randomly

to reduce Bais (underfitting)

Sequential learning

take high bias with low variance

Boosting

AdaBoost → exponential loss fun.

→ use week classifier (high bais low veriance)

→ used wieghted samples

→ used wieghted magority voting

s_1	0.1
s_2	0.6
x s_3	0.9
s_4	0.5

$\epsilon = \frac{0.9}{2.1}$

Steps:-

weight

① assign w.t to every sample ($w.t = \frac{1}{m}$)

② Train classifier C_1 (wieghted classifier + better than random)

→ compute w.ted error for $C_1 = \epsilon$

③ compute w.t of the classifier $= \frac{1}{2} \ln \left(\frac{1 - \text{error}}{\text{error}} \right)$ (ϵ, ϵ)

4. re w.t the samples αe^{-wt} , αe^{+wt} (reduce the correct, increase the incorrect)

origin wt

if it's correct

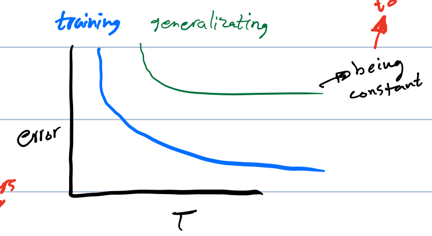
$$\alpha_i^{j+1} = \begin{cases} \alpha_i^j \cdot e^{-w.t}, & \text{if correct} \\ \alpha_i^j \cdot e^{+w.t}, & \text{if incorrect} \end{cases}$$

5. after multiple classifiers, we normalize the weights of samples

$$\alpha_i^{j+1} = \frac{\alpha_i^{j+1}}{\sum \alpha_k^{j+1}}$$

6. repeat 2-5 until 'T' time

※ of base learners "classifiers"



very robust to overfitting

$$\text{Predict} = \text{Sign}\left[\sum_{i=1}^{T} w_{T_i} f_i(x)\right]$$

of class

※ trick

$\sum$ Correct = 0.5

$\sum$ incorrect = 0.5

	1	2	
S_1	0.25	✗ 0.167	0.25
S_2	0.25	0.167	$\left(\frac{0.167}{0.167+0.5}\right)0.5$
✗ S_3	0.25	0.5	$\left(\frac{0.5}{1.67+0.5}\right)0.5$
S_4	0.25	✗ 0.167	0.25

※ Random Forest → Can be used for feature selection

Same as bagging (for samples)

- → randomly select features for each classifier $\sqrt{\#\text{ of original features}}$ (usually; this if the original is small)
- → base learner are DTs (decision trees)
- → hyperparameter = DT hyperparameter and T (※ of classifiers) and
- → harder intrupert than decision tree