→ **Deep Learning:** Learning Theory ✓ Finite hypothesis

Infinite hypothesis → VC-dimension

✓ $m \geq \frac{1}{\epsilon}\left(\ln|H| + \ln\left(\frac{1}{\delta}\right)\right)$

for finite hypothesis ω train error = 0

✓ $m \geq \frac{1}{2\epsilon^2}\left(\ln|H| + \ln\left(\frac{1}{\delta}\right)\right)$

finite hypothesis ω trainError ≠ 0

$|H| \to \infty$ Model complexity is computed using VC dimension of the hypothesis class.

VC-dimension: The maximum number of samples that can be $\boxed{\text{shattered}}$ by a hypothesis class is called VC-dimension.

... the ... complexity (VC)

Vapnik - Chervonenkis

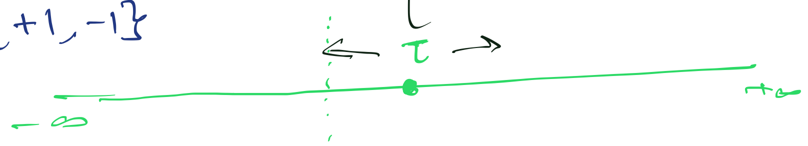
SHATTER: For all possible labellings of the samples, if there exist a hypothesis (model) 'h' with zero error (correctly classify all samples) in the hypothesis class 'H' is called shattering

Example:

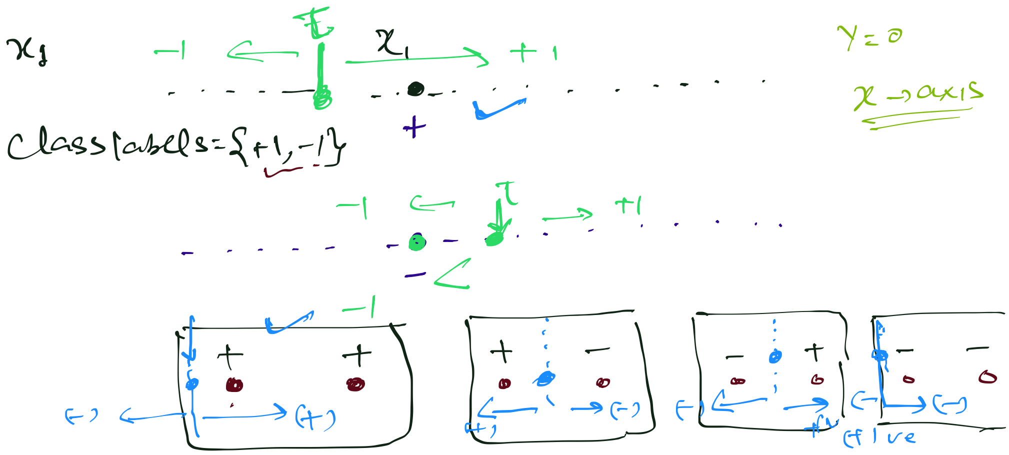
Let Threshold: τ $\begin{cases} +1 & \text{if } \tau \geq 0 \\ -1 & \text{otherwise} \end{cases}$

Class: $\{+1, -1\}$

Samples:



Class labels = {+1, -1}

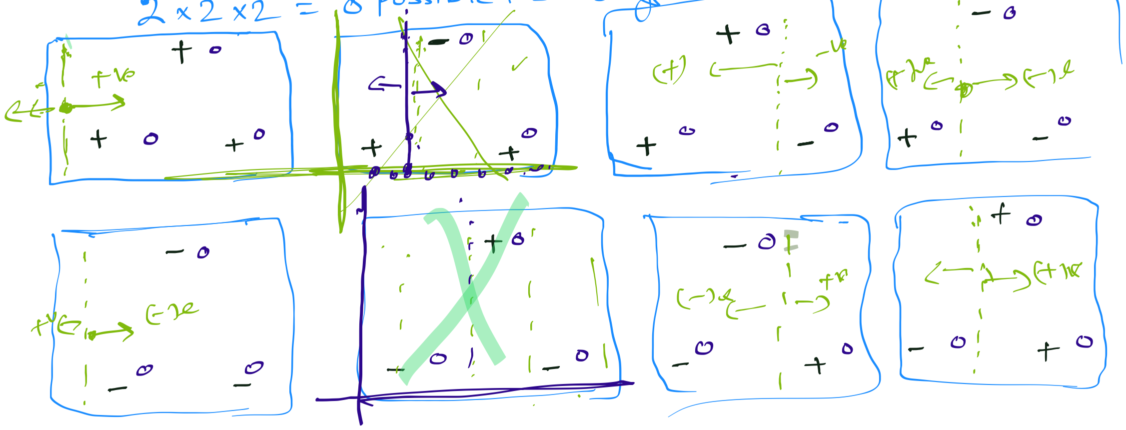


2 - Samples, 2 Classes :

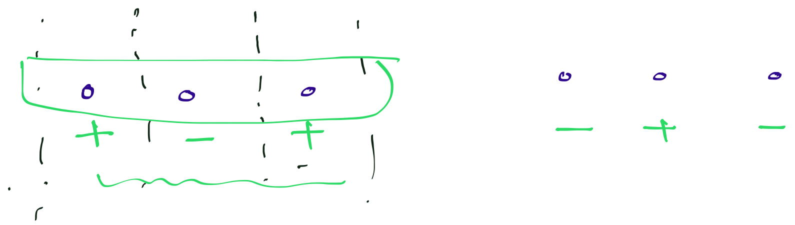
$2 \times 2 = 4$

3 - Sample, 2 Classes

$2 \times 2 \times 2 = 8$ possible lebellings



threshold does not exist which can shatter.



VC-dimension for threshold hypothesis class is 2. It can shatter a maximum of 2 samples.

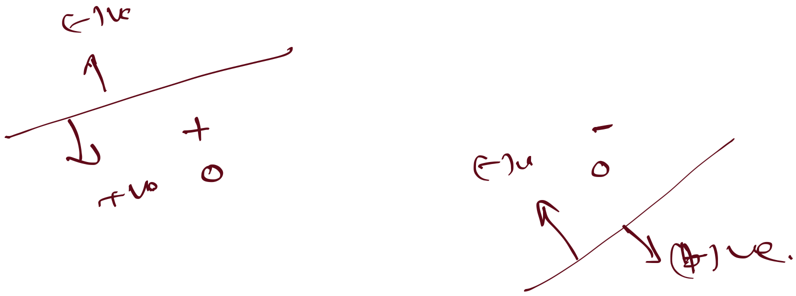
Example 2.

H: Linear models | Linear hyperplane.

Let example be H: linear model in 2D (line)

Class = {+1, −1}

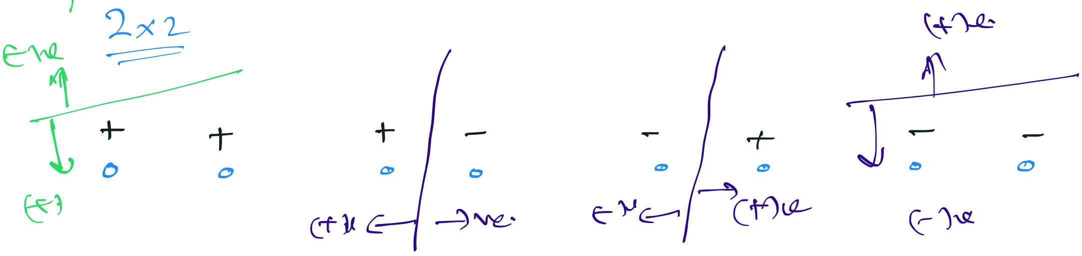
1. Sample:



✓

2 Samples, 2 Classes. $2^2 = 4$

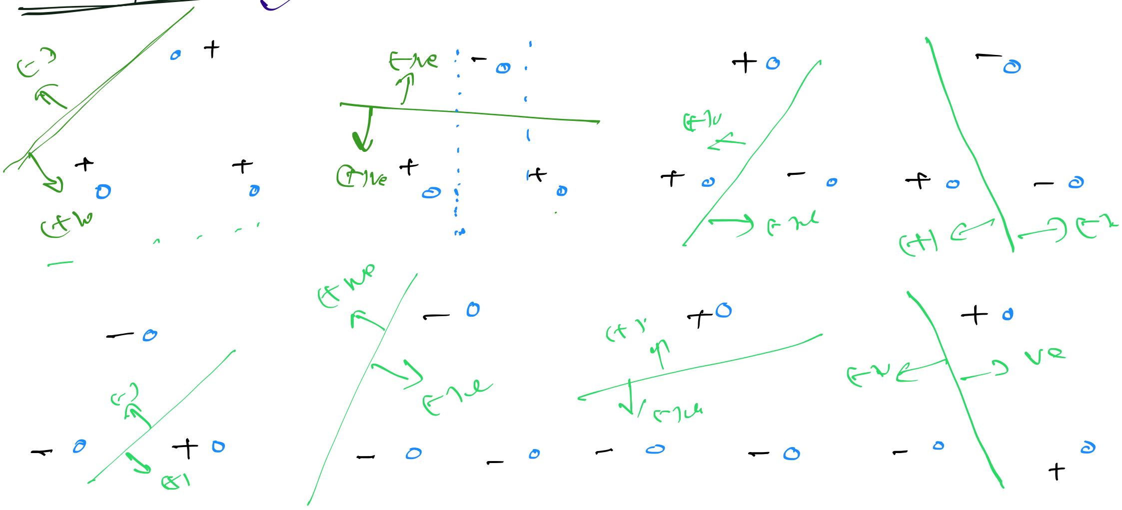
2×2

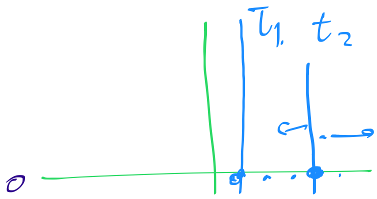


Line can Shatter 2 points.

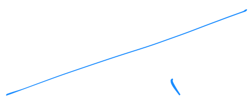
3-Samples, 2 classes:

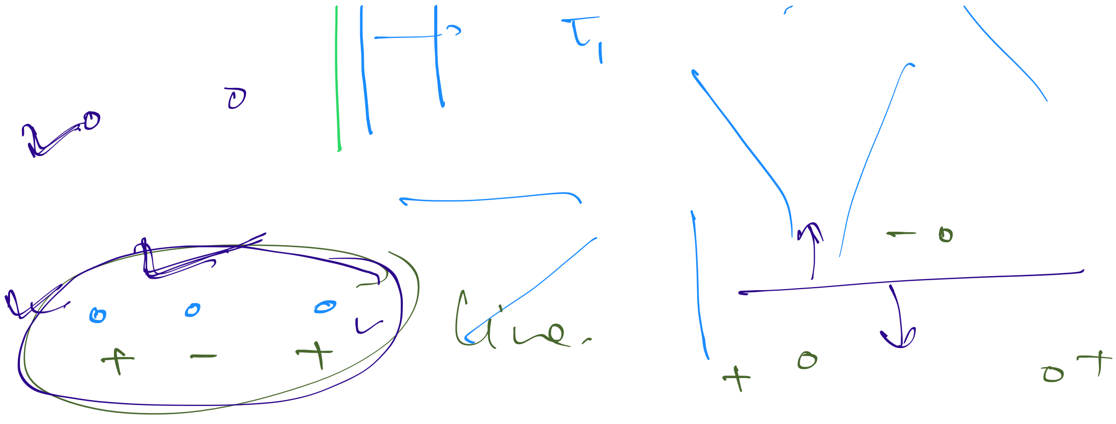
$2 \quad 2 \quad 2 \quad = 8$





$\leq \tau_c <$

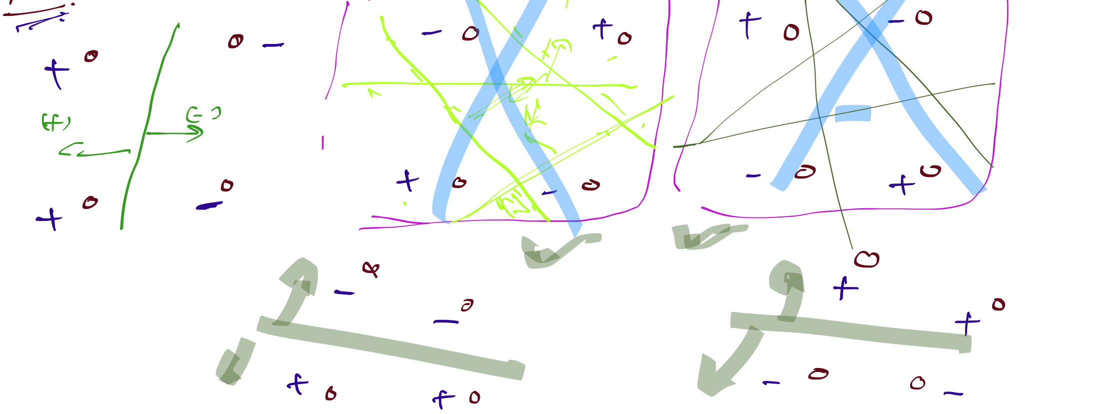




Line Can shatter 3 Samples.

4 Samples, 2 classes : $\underline{2} \times \underline{2} \times \underline{2} \times \underline{2} = 16$ Possible labellin.

Few:



Line can not shatter 4 - points

Line can shatter ~~at~~ maximum of 3-points.

VC-dimension (Line (or linear hypothesis in 2D)

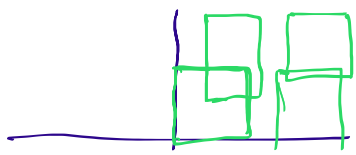
is $\underline{\underline{3}}$

VC-dimension of Linear hyperplane in d-dimension

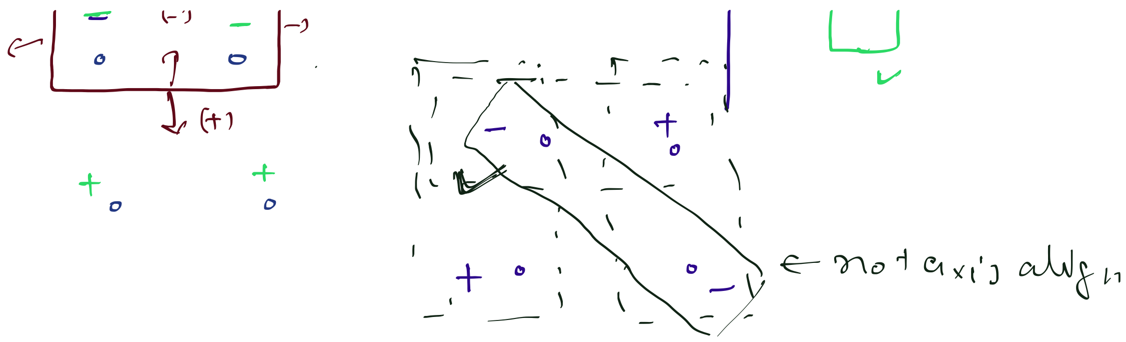
is $d+1$

Hypothesis is Axis-aligned <u>Rectange (2D)</u>

<u>4-Sample</u> class = 2



3

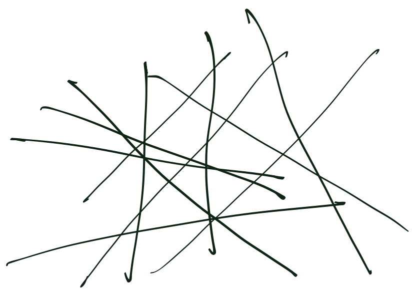


Rectangle in 2D → VC-dim = (4)

H = Line in 2D →

$$y = mx + c$$

$$\boxed{w^T x_i + b = 0}$$



- ✓ b: bias (Distance of the plane from origin of the vector space)
- ✓ w: weight (Vector perpendicular to the plane having x_i)

In a d-dimension — i^{th} Sample x_i

$VC(h) \propto$ # parameter

$$x_i = \begin{bmatrix} x_i^1 \\ x_i^2 \\ \vdots \\ x_i^d \end{bmatrix}_{d\times 1} ; \quad w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix} ; \quad b \in \mathbb{R}$$

$$h_{w,b}(x_i) = w^T x_i + b$$

$$h_{w,b}(x_i) = \boxed{w^T x_i + b = 0}$$

$x_i \in \mathbb{R}^d$ $\quad$ $w \in \mathbb{R}^d$

$VC(h) = d+1$

Linear model in d-dimension.

$m \geq O(VC(H))$	$m \geq O\left(\frac{1}{\delta}\right)$
	$\boxed{\simeq m \geq O((1-\delta))}$
$m \geq O(VC(H) \lg VC(H))$	$m \geq O\left(\frac{1}{\epsilon}\right)$
	Zero True
$m \geq O(VC(H)^2)$	$m \geq O\left(\frac{1}{\epsilon^2}\right)$
Structure	$\epsilon_{tr} \neq 0$
	$m \geq O(\ln\lvert H\rvert)$

m: # samples

~~Total~~

ϵ_{10} / ~~generalization~~

(error | Tolerance)

True Error / Generallization Error

$= \text{Train Error} + \epsilon$

Structural Error

Finite Hyp

(i) $m \geq \frac{1}{\epsilon}\left(\ln(|H|) + \ln\left(\frac{1}{\delta}\right)\right)$ # Train Error = 0

(ii) $m \geq \frac{1}{2\epsilon^2}\left(\ln|H| + \ln\left(\frac{1}{\delta}\right)\right)$ # Train Err ≠ 0

(iii) $m \geq \frac{1}{\epsilon^2}\left(C_1 . VC(H) + C_2 . \ln\left(\frac{C_3}{\delta}\right)\right)$

C_1, C_2, C_3 → Constant

(i) $\epsilon \le \dfrac{\ln(|H|) + \ln(\frac{1}{\delta})}{m}$

$\epsilon \propto (1-\delta)$

Prob Conf

(ii) $\epsilon = \sqrt{\dfrac{\ln(|H|) + \ln(\frac{1}{\delta})}{2m}}$

(iii) $\epsilon \simeq \sqrt{\dfrac{c_1 \cdot VC(H) + c_2 \ln(c_3/\delta)}{m}}$

$(1-\delta)$ probabi

m

δ

Realtime Error

empirical error

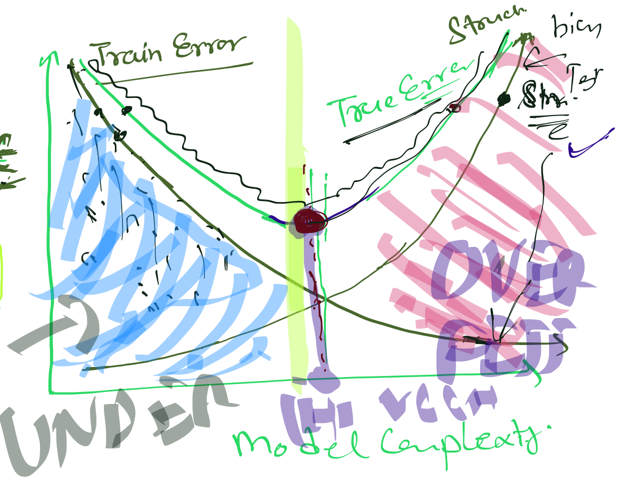
Structure err

ϵ

True Error $\le$ Train Error $+\ \epsilon$

True is Error is high

Train Error is low



overfitting (Variance)   VC(H) < |H|

Train Error → high : Underfitting. (bias problem)

Good hypothesis : Low overfittin Low Under fitt.

high Variance → overfitting.
high bias → Underfitting

Bias - Variance Trade off

Model : low bias & low variance is a good

→ Model is suffering from overfitting (variance) issue:

Remedies :- Reducing the model complexity
↳ Regularization
↳ Dropout
↳ Change to less complex model

$\epsilon \propto \ln(\frac{1}{\delta})$

(ϵ → reduces)

- Increase the # samples in the training
  ↳ Augmentation (Gen methods)
- NN: (early stopping)
- "δ": → ~ $(1-\delta)$ reduced. → High Confidence

$P(\text{True Error} \leq \epsilon) > (1-\delta)$ Conf.

(True $=$ ...)

$\delta = 0.01$

$1-\delta = .99$

$\delta = 0.05$

$1-\delta = .95$

Confidence interval, p value
{ Significance
Null hypothesis

→ Distribution Shift:

test

Training data, Validation data, Test data

If there is a mismatch (all these three does n't follow same distribution: distribution shift)

↳ Collecting data from different sources

Validation ↳ Data splitting during training

(upto 1000)

↳ Cross-Validation (

Small-data: Leave-one-out (LOOCV)

medium data: 80-20, 70-30, 60-40, (Train, validat) + k-fold Cross-Valt.

Large dataset: 1 to 10% data for validation

↳ low k-fold (k = 3 to 5)

↳ Random sampling for fixing the hyper-parameters.

Hyper-parameters: Set by the user.

(Cross-validation, Splitting)

Validation

Step 1: Split data into 2-parts → Training | Testing

Step 2: Training data → (Train' + Validation)

Step 3: Fit the model with hyperparameters with good validation accuracy. (CV, splitting)

Step 4: Fixed Hyperparameters in the previous step

Train the model on whole Training Data and

Step 5: Evaluate model on Test Data

Increase the validation % ge.

Real T.

	Train error	Validation	Test error
M_1 :	0.1 (10%)	0.12 (12%)	0.4 (40%)

Distribution Shift is found:

Normalization of feature

. Try to Increase the %ge of validation

Random Sample the samples from Test Distribution and Try to finetune the large model on this random samples. (If you are not able to retrain)

Otherwise Augment the Training Data with random samples from Test distribution and retrain the model and fix the hyper-parameter using validation data

→ Model is suffering from Underfitting (bias)

Issue: (Training error is high)

Biased Data

Hand

Winter

Blurry Image

- ↳ Increase the model Complexity
  - ↳ Somehow try to increase the VC-dim
    - ↳ Introducing non-linearity
  - ↳ Change to complex model.
  - ↳ Increase the epoch/iterations
- ↳ Look into the Data
  - ↳ Preprocessing (Reduce lossy technique)
  - → Enhancement

Example:

	train Err	Validation Error #s	Test Error #s	Problem in any
M1	10%	9%	11%	Low bias, low var. No dist shift
M2	8%	20%	18%	Low bias, High variance, ??
M3	30%	40%	45%	High bias, ?? ??
M4	13%	14%	20%	Low bias, Low variance, Dist. Shift Issue

+3.5%

overfitting

→ . These remedies doesn't guarantee the perfect model :

- In practice It works
- medium size
  Moderate size data is needed $\geq 10k$ samples
  $\epsilon \propto O\left(\frac{1}{m}\right)$: PAC learn.
- Experience Matters (Expert)

+ Comparing the models:: Statistical Test are required
  errors: $(\bar{\mu}_{m_1} \pm CI_{m_1}), (\mu_{m_2} \pm CI_{m_2})$ ...... ( )

Model with low mean error and low confidence Interval on error is a GOOD model

---

i^{th} model: Repeat the training 'n' times with different seeds.
n
times
[n ≠]

$$\mu_{m_i} = \frac{1}{n} \cdot \sum_{j=1} \underline{\text{validateTest}_j \cdot (e_j)}$$

$$CI_{m_i} = \frac{\sigma^2}{\sqrt{n}} \leftarrow \sqrt{\text{variance}}$$

e_1

$\vdots$

e_n n

Testcn