SUPERVISED LEARNING:

GOAL: LEARN A PREDICTOR $f_\theta : x \to y$

WHERE $x \in \mathcal{X}$ INPUTS

$y \in \mathcal{Y}$ TARGETS

f_θ IS THE PREDICTOR WITH TRAINABLE WEIGHTS $\theta \in \mathbb{R}^p$

INPUTS $\mathcal{X}$:

- IMAGES: $\mathcal{X} = [0,1]^{3n}$, n RGB PIXELS
- TIME SERIES: $\mathcal{X} = [0,T]$
- TEXT
- GRAPHS

TARGETS $\mathcal{Y}$:

(1) CLASSIFICATION

$\mathcal{Y} = \{0, 1, \dots, C\}$ CLASS LABELS

(2) REGRESSION

$\mathcal{Y} = \mathbb{R}$ or $\mathcal{Y} = \mathbb{R}^n$

RISK MINIMIZATION:

$$\mathcal{R} = \mathbb{E}_{x,y}\{\mathcal{L}(f_\theta(x), y)\}$$ RISK

WHERE $\mathcal{L}: \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$ IS A LOSS FUNCTION.

IN PRACTICE, WE ONLY HAVE ACCESS TO A SUPERVISED DATASET $\mathcal{D} = \{(x_i, y_i)\}_{i=1,\dots,N}$

$$\hat{\mathcal{R}}(\theta) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}(f_\theta(x_i), y_i)$$ EMPIRICAL RISK

EMPIRICAL RISK MINIMIZATION (ERM):

$$\hat{\theta} = \arg\min_\theta \mathcal{R}(\theta)$$

IDEA: FIND THE PARAMETERS THAT MINIMIZE THE EMPIRICAL RISK

IMPORTANT QUESTIONS:

(1) HOW DO WE CHOOSE λ FOR OUR TASK?

(2) HOW DO WE PARAMETRIZE $f_\theta(x)$?

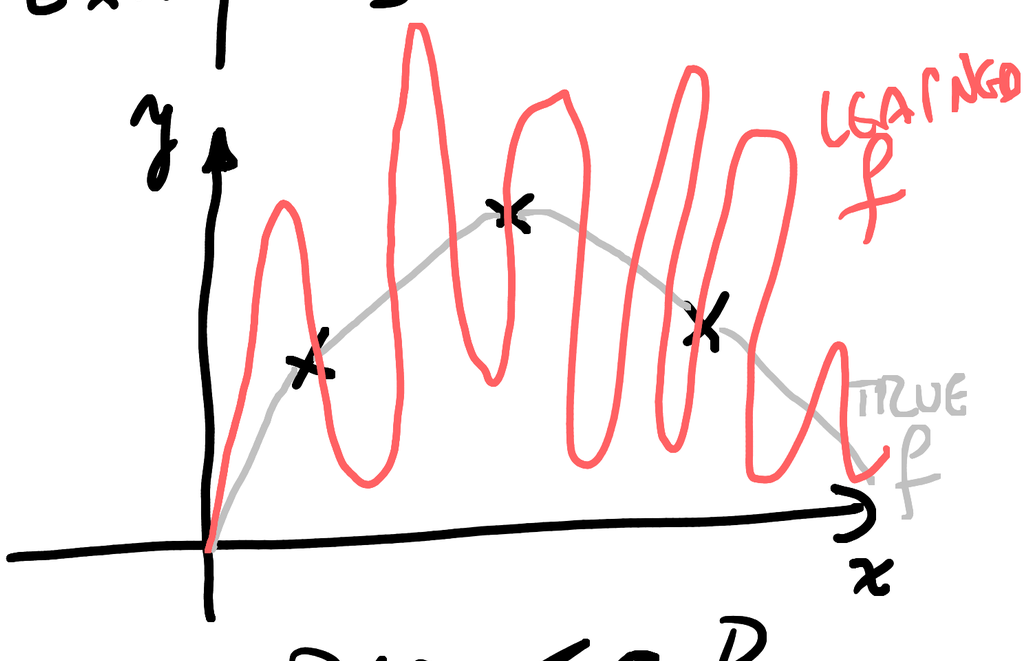
(3) HOW DO WE OPTIMIZE THE EMPIRICAL RISK?

(4) MINIMIZING $\hat{R}(\theta)$ = MINIMIZING $R(\theta)$?

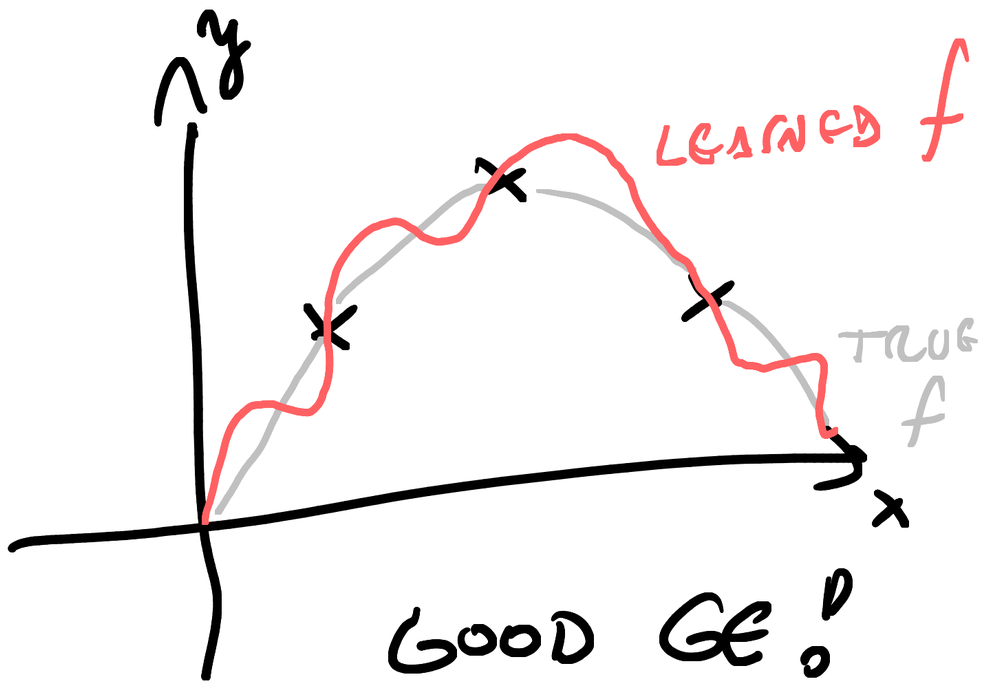
GENERALIZATION ERROR

$$GE = |R - \hat{R}|$$

EXAMPLES x = TRAINING POINTS



BAD GE!



GOOD GE!

NOTE THAT IN BOTH CASES WE HAVE $\hat{R} = 0$ ZERO EMPIRICAL RISK

OPTIMIZATION:

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} L(f_\theta(x_i), y_i)$$

OPTIMALITY CONDITION (ASSUMING DIFFERENTIABILITY)

$$\sum_{i=1}^{N} \frac{dL}{d\theta}(\theta^*) = 0$$

GRADIENT DESCENT

$$\theta^{K+1} = \theta^K - \eta \frac{dL}{d\theta}(\theta^K)$$

η STEP SIZE OR LEARNING RATE.

(MORE ON THIS IN NELLY'S CLASS)

CHOOSING THE LOSS $\mathcal{L}$

① BINARY CLASSIFICATION

- TWO CLASSES $\mathcal{Y} = \{-1, 1\}$

0-1 LOSS

$$\mathcal{L}^{01}(\hat{y}, y) = \begin{cases} 1 & \text{if } \hat{y} \neq y \\ -1 & \text{if } y = \hat{y} \end{cases}.$$

THEOREM :

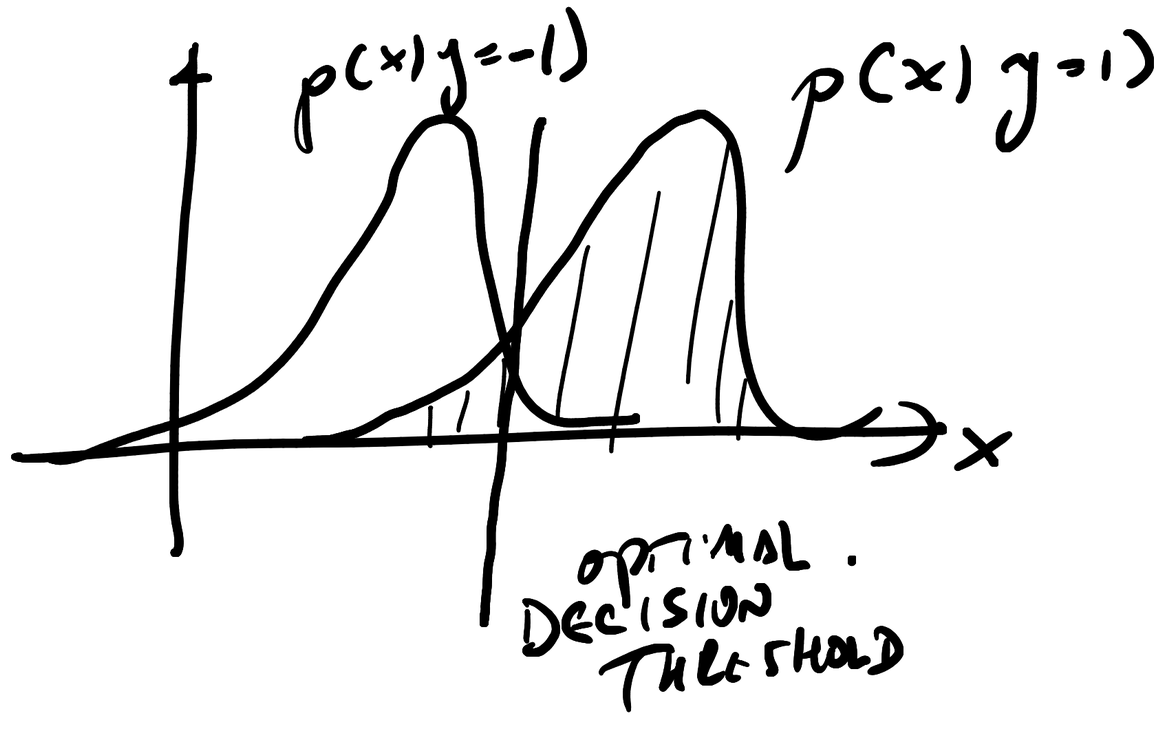
$$\hat{f} = \underset{f}{\operatorname{argmin}} \; \mathbb{E}\left\{ \mathcal{L}^{01}(f(x), y) \right\}$$

$$\hat{f}(x) = \begin{cases} -1 & \text{if } p(y=-1/x) \geq p(y=1/x) \\ 1 & \text{if } p(y=-1/x) < p(y=1/x) \end{cases}$$

THUS WE NEED TO CHECK THE RATIO

$$\frac{p(y=-1/x)}{p(y=1/x)} = \underbrace{\frac{p(x/y=-1)}{p(x/y=1)}} \frac{p(y=-1)}{p(y=1)}$$

LIKELIHOOD RATIO TEST.

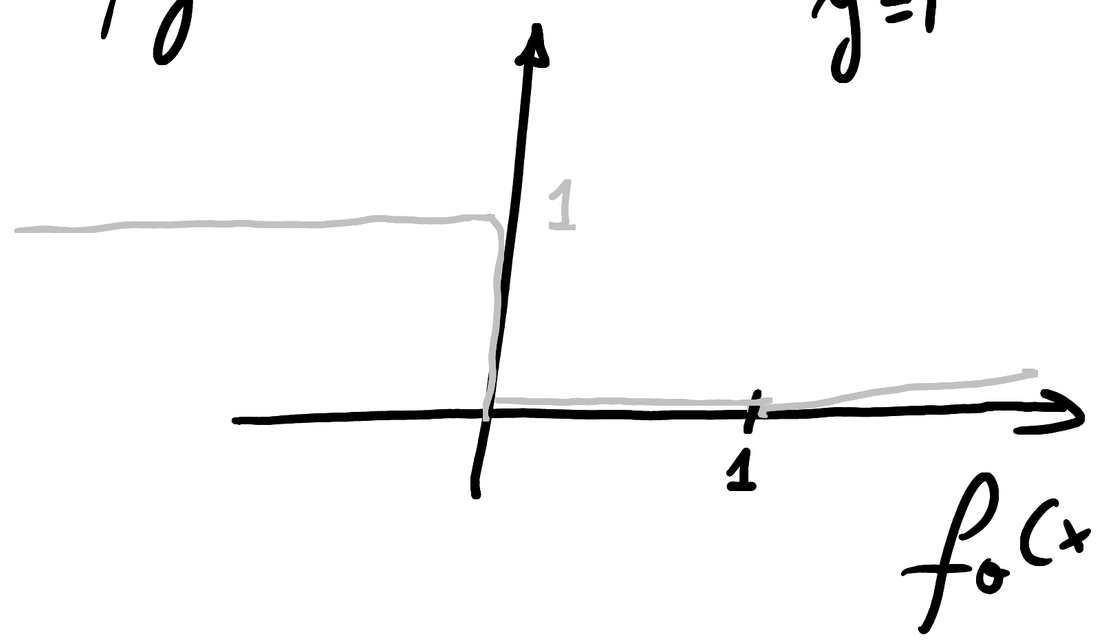


HOWEVER IN PRACTICE WE DON'T HAVE ACCESS TO $p(x/y)$ .... ONLY A DATASET $\{(x_i, y_i)\}_i$

IDEA : LEARN A FUNCTION $f_\theta(x) : \mathcal{X} \to \mathbb{R}$
AND CHOOSE CLASS AS $\text{sign}(f_\theta(x))$.

0-1 Loss

$\mathcal{L}(\text{sign} f_\theta(x), y)$



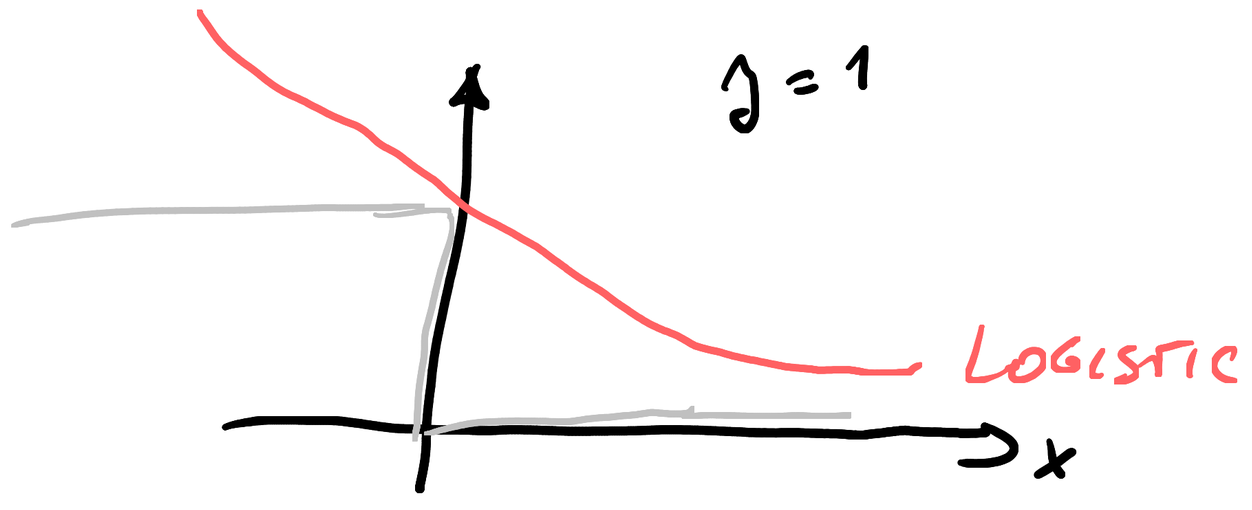
problem $\Rightarrow$ THIS LOSS HAS ZERO GRADIENTS ALMOST EVERY WHERE ...

$\downarrow$

SURROGATE LOSSES

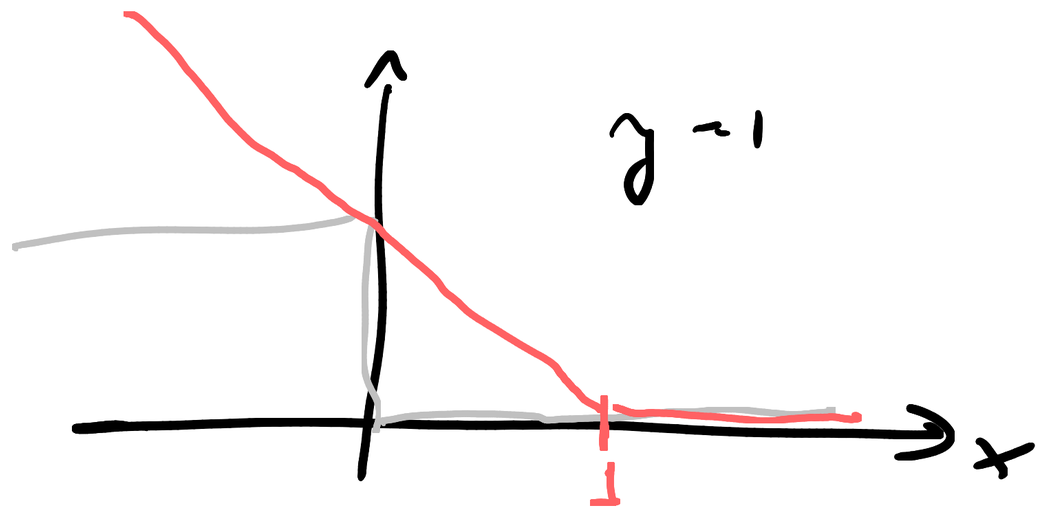
LOGISTIC LOSS:

$$\mathcal{L}(f_\theta(x), y) = \log\left(1 + e^{-y f_\theta(x)}\right)$$



HINGE LOSS:

$$\mathcal{L}(f_\theta(x), y) = \max(0, 1 - y f_\theta(x))$$



(2) REGRESSION

QUADRATIC LOSS

$$\mathcal{L}(f_\theta(x), y) = \| f_\theta(x) - y \|^2$$

THEOREM: optimal predictor

$$\hat{f} = \arg\min_f \mathbb{E}_{x,y}\left\{ \| f_\theta(x) - y \|^2 \right\}$$

$$\Rightarrow \hat{f}(x) = \mathbb{E}\{ y \mid x \}$$

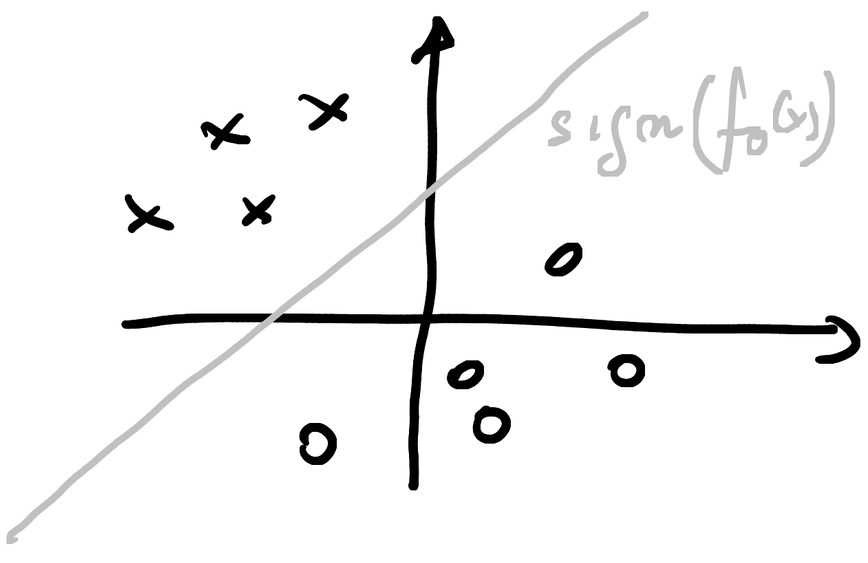
KNOWN AS "POSTERIOR MEAN"

Choosing the predictor f_θ

LINEAR predictor $f_\theta(x) = \theta_1^T x + \theta_2$ $\quad \theta = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}$

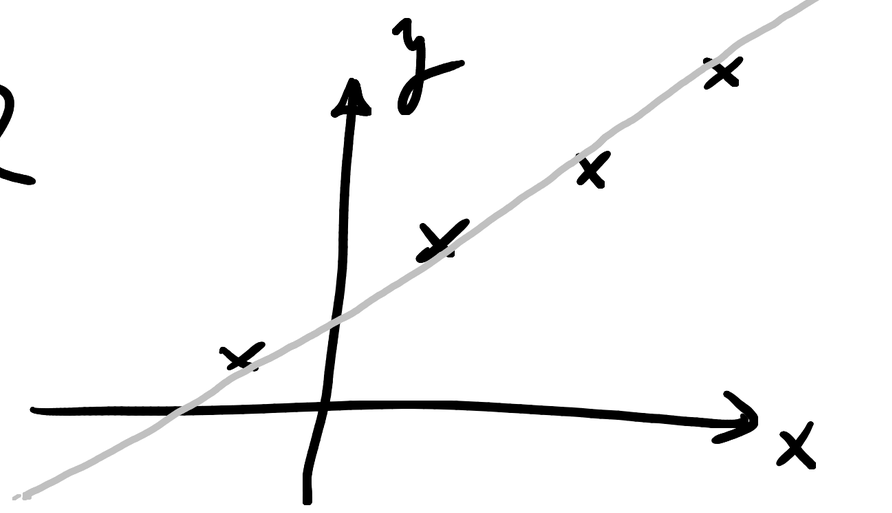
CLASSIFICATION: LINEAR BOUNDARY $\theta_1^T x > \theta_2$

$\mathcal{X} = \mathbb{R}^2$



REGRESSION: LINEAR FUNCTION

$\mathcal{X} = \mathbb{R}$



SIMPLE SOLUTION if $\mathcal{L}$ = QUADRATIC LOSS

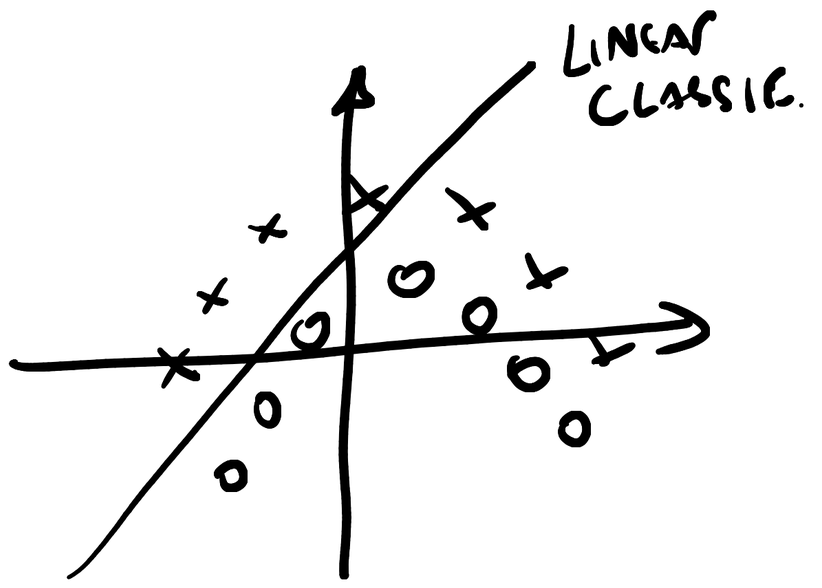
$$\hat{\theta} \in \arg\min_\theta \sum_{i=1}^N \| \theta^T x_i - y_i \|^2 \iff \arg\min_\theta \| X\theta - \bar{y} \|^2$$

$X = [x_1 \dots x_N]$ $\quad N \geq m$ $\quad \bar{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}$

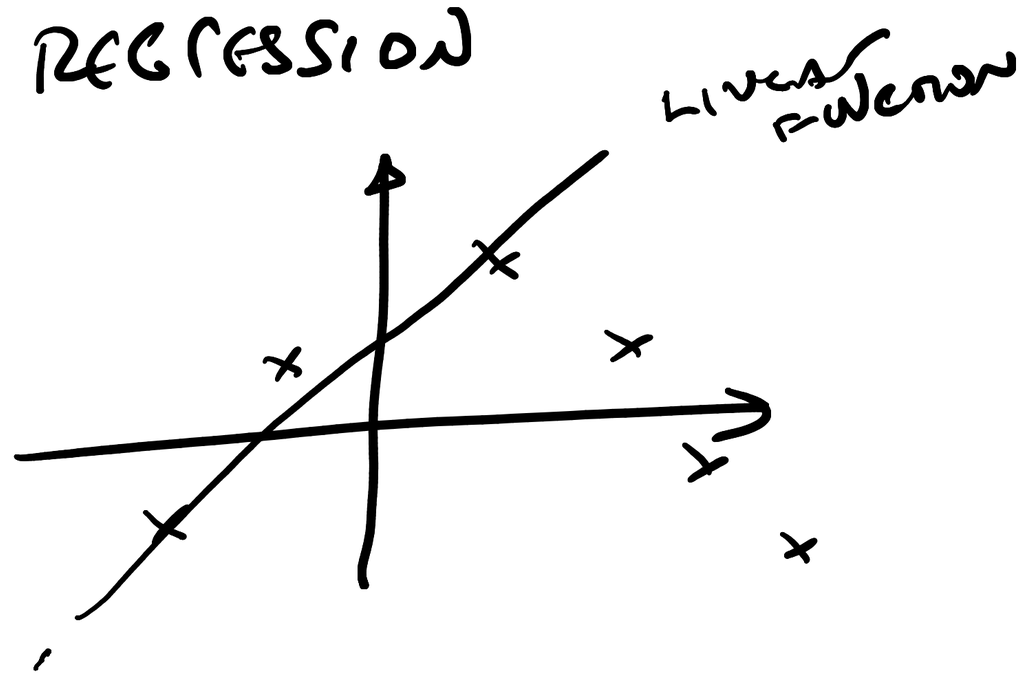
if $N > m$ $\quad \hat{\theta} = (X^T X)^{-1} X^T \bar{y}$

LIMITATIONS OF LINEAR REGRESSION:

CLASSIFICATION



REGRESSION



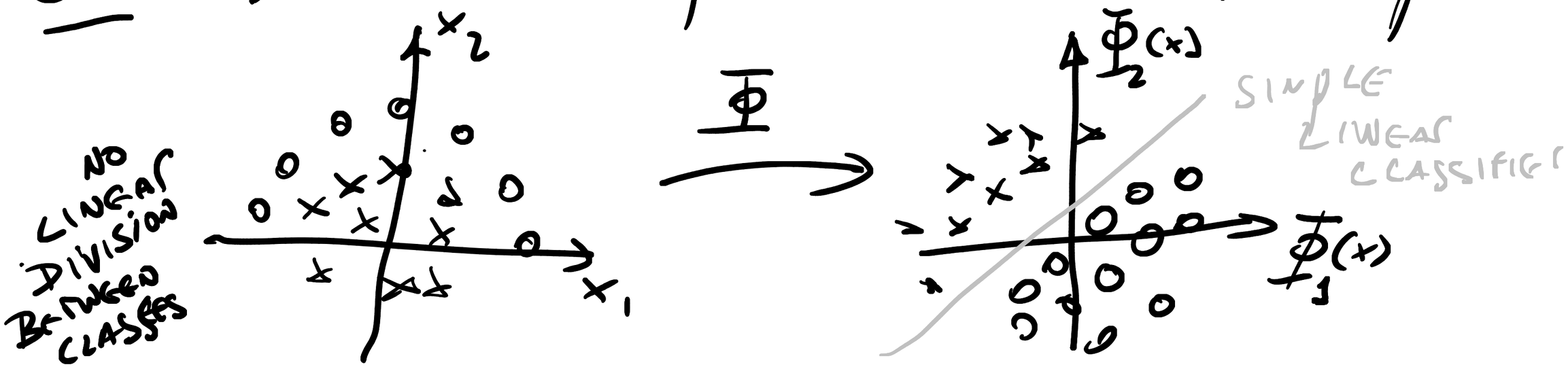
LIFTING

$$f_\theta(x) = \theta^T \Phi(x)$$

where $\Phi: \mathcal{X} \to \mathbb{R}^p$ is CALLED THE FEATURE MAPPING

- IN A WAY, WE REPLACE DATASET $\{(x_i, y_i)\}$ by $\{(\Phi(x_i), y_i)\}$
- f_θ IS STILL LINEAR IN θ!

GOAL: LINEARIZE THE PROBLEM IN THE LIFTED SPACE.



POLYNOMIAL Φ (order s)

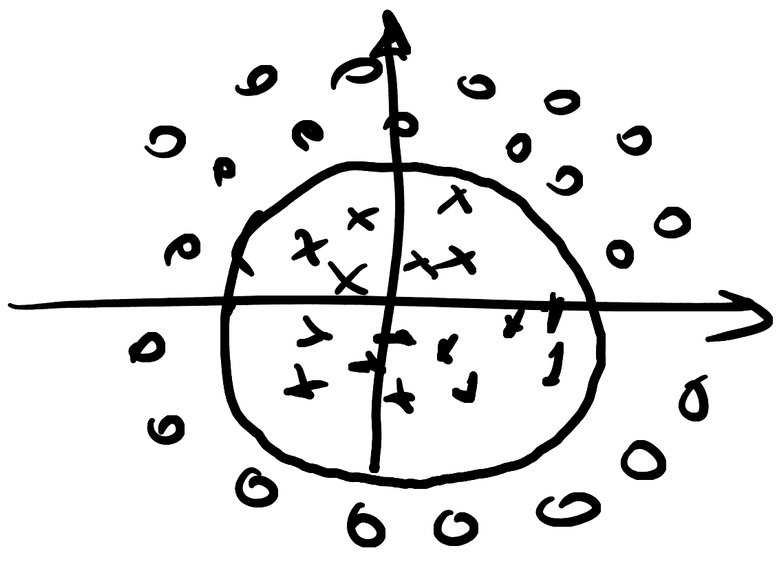
$\Phi(x)$ CONTAINS ALL THE POLYNOMIALS UP TO ORDER S

EXAMPLE

$\mathcal{X} = \mathbb{R}^2 \qquad x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

$\Phi(x) = \begin{bmatrix} x_1 \\ x_2 \\ x_1 x_2 \\ x_1^2 + x_2^2 \end{bmatrix} \qquad s = 2$

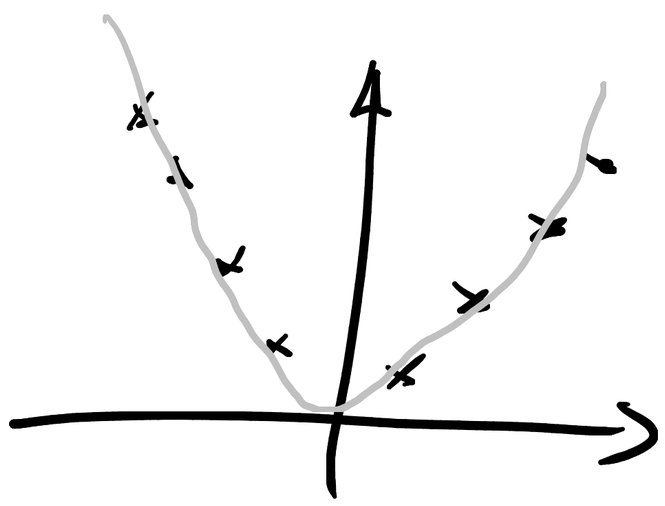
CLASSIFICATION



CHOOSE

$\theta = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \Rightarrow f_\theta(x) = x_1^2 + x_2^2$

REGRESSION WITH $\mathcal{X} = \mathbb{R}$



$\Phi(x) = \begin{bmatrix} x_1 \\ x_1^2 \end{bmatrix}$

CHOOSE

$\theta = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \Rightarrow f_\theta(x) = x^2$

HOW MANY FEATURES DO YOU NEED?

- MORE FEATURES → HIGHER DIMENTIONAL SPACE

↓

MORE CHANCES OF LINEARIZATION

A REGRESSION ANALYSIS:

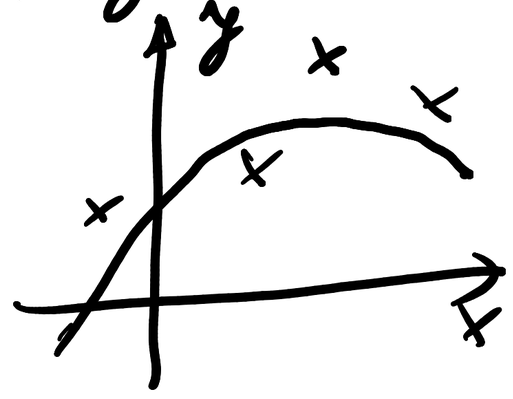
$$\mathcal{X} = \mathbb{R}^n$$

ERM $\quad \arg\min_{\theta} \|\Phi^T\theta - \vec{y}\|^2 \quad$ where $\Phi = [\Phi(x_1) \dots \Phi(x_N)]$ $\quad p \times N$

if $N > p$ UNDERPARAMETERIZED

$$\Rightarrow \hat{\theta} = (\Phi\Phi^T)^{-1}\Phi y$$ UNIQUE SOLUTION

IN GENERAL $f_{\hat{\theta}}(x_i) \neq y_i$

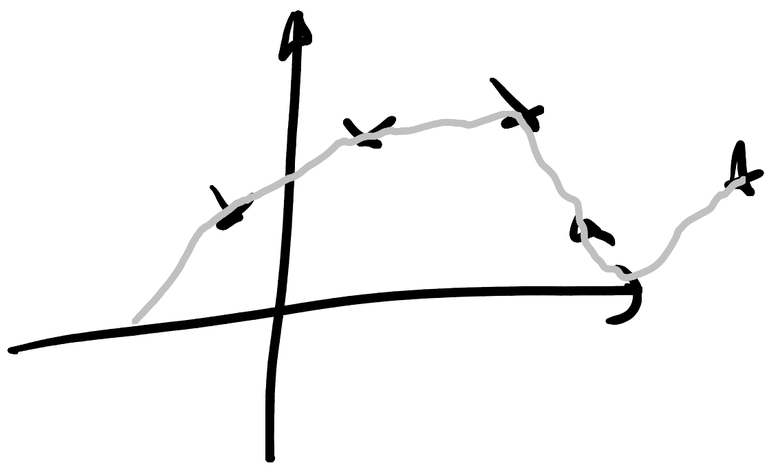


if $N \leq p$ OVERPARAMETERIZED

- THERE ARE INFINITE SOLUTIONS TO THE ERM problem
- ALL THE SOLUTIONS VERIFY PERFECTLY FIT DATASET $f_{\hat{\theta}}(x_i) = y_i \quad \forall i = 1, \dots, N$

$\Leftrightarrow \hat{R}(\hat{\theta}) = 0$

INTERPOLATING SOLUTION



Example

$\hat{\theta} = \Phi (\Phi^T \Phi)^{-1} \bar{y}$ IS ONE SOLUTION (CALLED MINIMUM NORM SOLUTION).

$\| \Phi^T \hat{\theta} - \bar{y} \|^2$

$= \| \Phi^T \Phi (\Phi^T \Phi)^{-1} \bar{y} - \bar{y} \|^2 = \| \bar{y} - \bar{y} \| = 0$

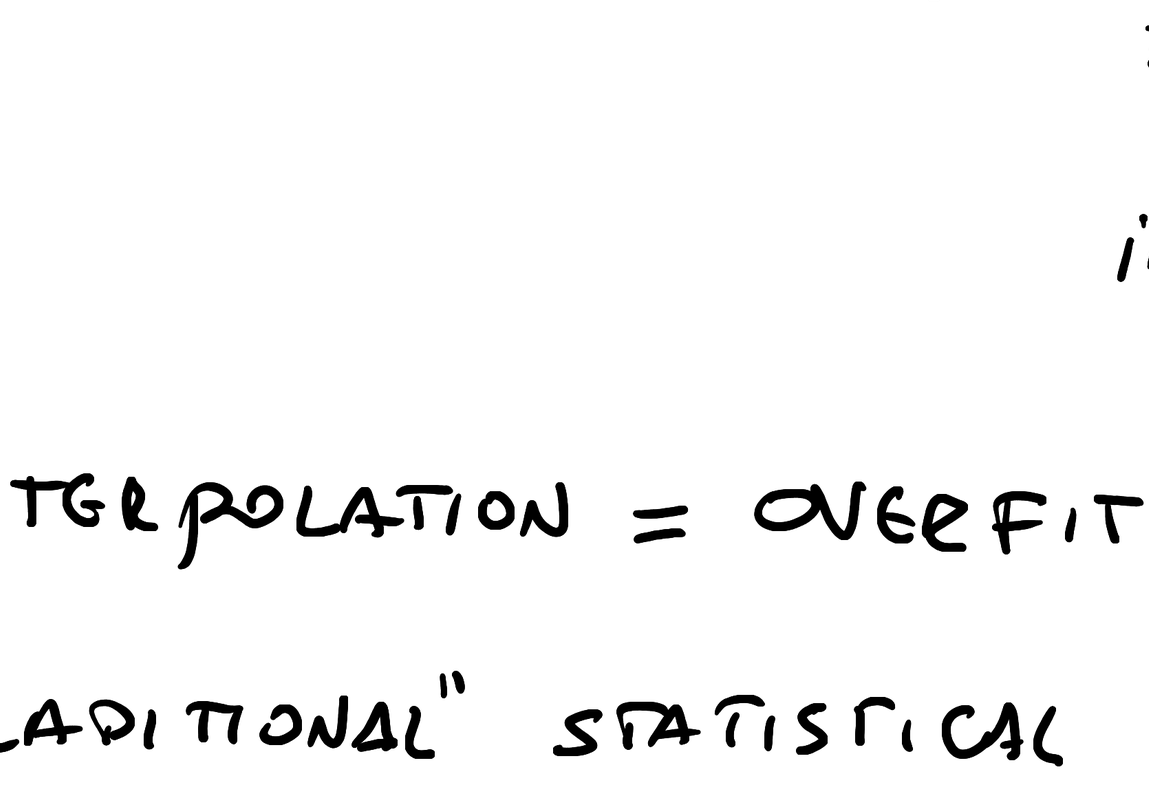
ZERO EMPIRICAL RISK

$=$

INTERPOLATING SOLUTION.

INTERPOLATION = OVERFITTING?

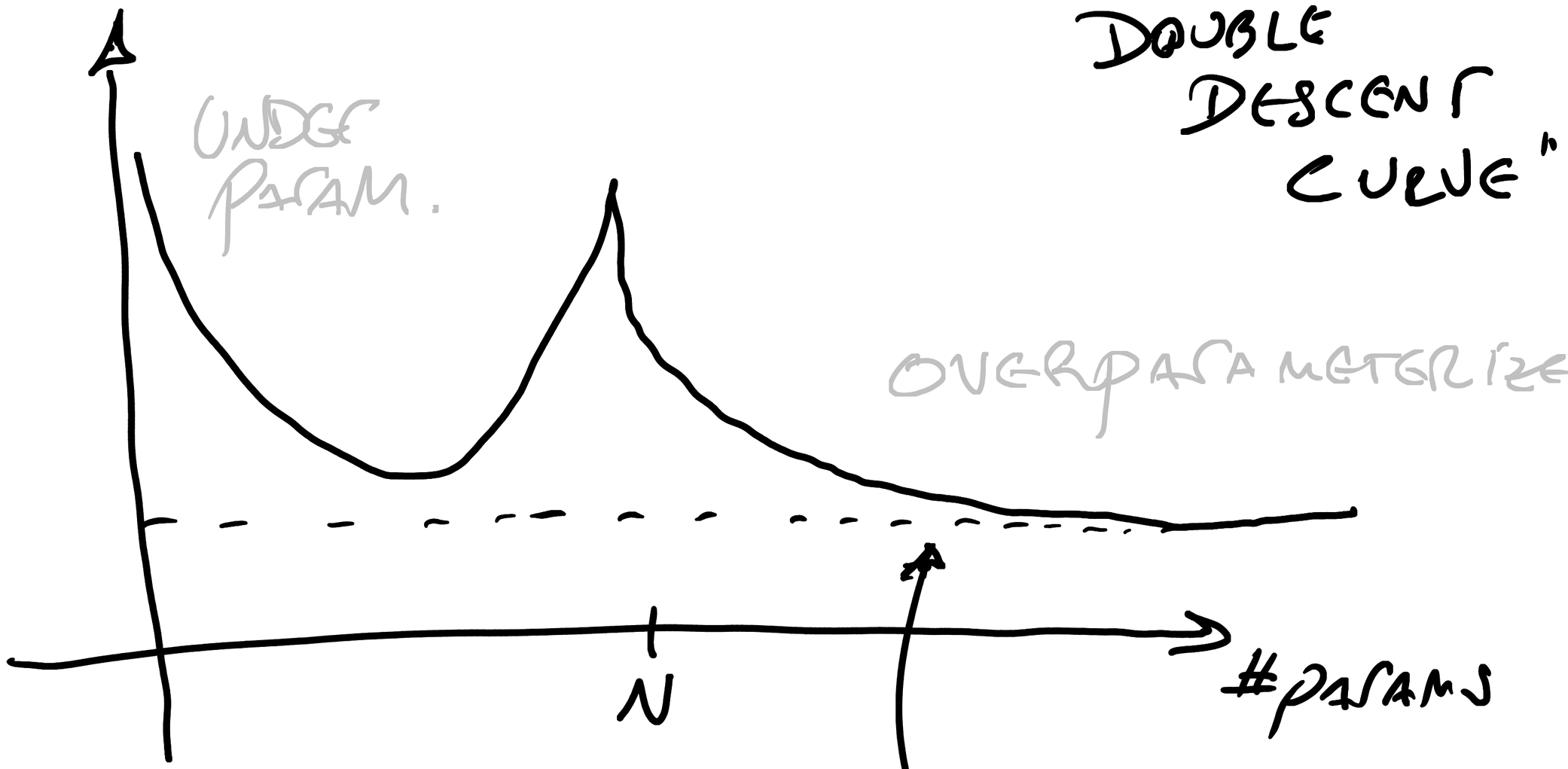
"TRADITIONAL" STATISTICAL BIAS VS VARIANCE



VON NEUMANN:

"WITH 4 PARAMETERS I CAN FIT AN ELEPHANT".

MODERN MACHINE LEARNING



"DOUBLE DESCENT CURVE"

BASIS FUNCTIONS

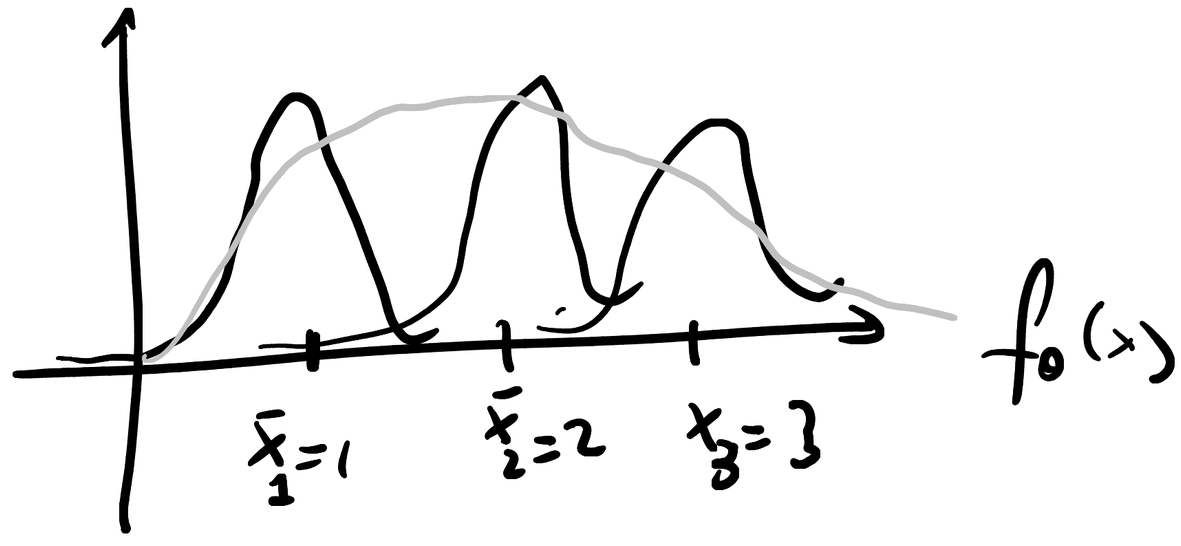
$$[\Phi(x)]_j = \omega(\|x - \bar{x}_j\|)$$ where

$\bar{x}_j$ is a KNOT POINT

EXAMPLE: RADIAL BASIS FUNCTIONS (RBFs)

$$\omega(t) = e^{-\frac{t^2}{\sigma^2}}$$

$$\Rightarrow f_\theta(x) = \sum_{j=1}^{p} \theta_j \, e^{-\frac{\|x - \bar{x}_j\|^2}{\sigma^2}}$$



- HOW SHOULD WE CHOOSE $\bar{x}_j$ IN HIGH DIMENSIONS?

KERNELS:

If $p >> N$, COMPUTING THE EXPLICIT FEATURE MAP MIGHT BE COSTLY! HOWEVER WE SAW THAT:

$$\hat{\theta} = \bar{\Phi}\left(\bar{\Phi}^T\bar{\Phi}\right)^{-1}\bar{y}$$

AND $$\hat{f}_\theta(x) = \hat{\theta}^T\bar{\Phi}(x) = \bar{y}^T\left(\bar{\Phi}^T\bar{\Phi}\right)^{-1}\bar{\Phi}^T\bar{\Phi}(x)$$

DEFINE $K = \bar{\Phi}^T\bar{\Phi}$ GRAM MATRIX

$$K_{ij} = \bar{\Phi}(x_i)^T\bar{\Phi}(x_j)$$

$$\triangleq k(x_i, x_j)$$ KERNEL FUNCTION

$$\Rightarrow \hat{f}_\theta(x) = \bar{y}^T K^{-1}\begin{bmatrix} k(x_1, x) \\ \vdots \\ k(x_n, x) \end{bmatrix}$$

SINCE $$\bar{\Phi}^T\bar{\Phi}(x) = \begin{bmatrix} \bar{\Phi}(x_1)^T\bar{\Phi}(x) \\ \vdots \\ \bar{\Phi}(x_n)^T\bar{\Phi}(x) \end{bmatrix} = \begin{bmatrix} k(x_1, x) \\ \vdots \\ k(x_n, x) \end{bmatrix}$$

OBSERVATIONS:

① OFTEN, COMPUTING $K(x_i, x_j)$ IS CHEAPER THAN $\Phi(x_i)^T \Phi(x_j)$!

FOR EXAMPLE:

a) $\Phi(x) = \begin{bmatrix} 1 \\ \sqrt{2}x \\ x^2 \end{bmatrix} \iff K(x,u) = (1+xu)^2$

b) $K(x,u) = e^{-\frac{(x-u)^2}{\sigma^2}} = e^{-\frac{x^2}{\sigma^2}} e^{-\frac{u^2}{\sigma^2}} e^{\frac{2xu}{\sigma^2}}$

$(x, u \in \mathbb{R})$

$$= \sum_{j=0}^{\infty} \frac{(2xu)^j}{j!} \cdot e^{-\frac{x^2}{\sigma^2}} e^{-\frac{u^2}{\sigma^2}}$$

$$= \Phi(x)^T \Phi(u)$$

where $[\Phi(x)]_j = e^{-\frac{x^2}{\sigma^2}} \frac{\sqrt{2}x^j}{j!}$

HERE $p = \infty$ INFINITELY MANY FEATURES!

↓

IT WOULD BE IMPOSSIBLE TO COMPUTE THE FEATURE MAP EXPLICITLY.

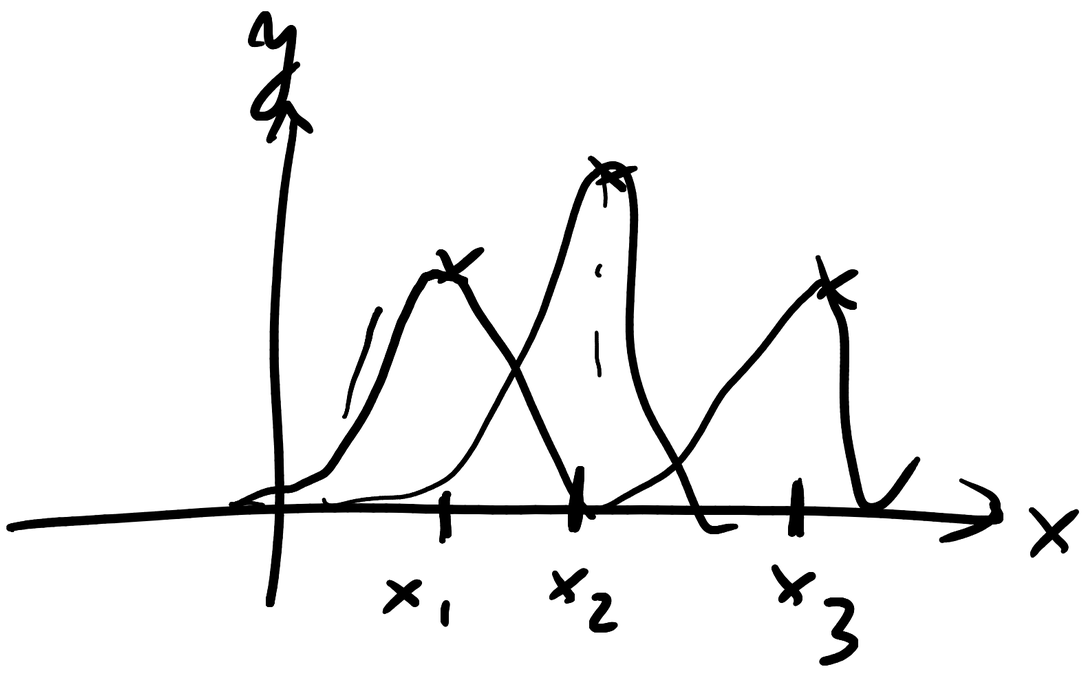
(2) KNOT POINTS

SINCE $\hat{f}_\theta(x) = \bar{y}^T K^{-1} \begin{bmatrix} k(x_1, x) \\ \vdots \\ k(x_m, x) \end{bmatrix}$

THE SOLUTION ALWAYS HAS THIS FORM. → $\hat{f}_\theta(x) = \sum_{i=1}^{N} \alpha_i K(x_i, x)$ REPRESENTER THEOREM

where $\alpha_i = [y^T K^{-1}]_i$

WE CAN SEE $K(x_i, x)$ AS A BASIS FUNCTION WITH KNOT POINTS $\bar{x}_i = x_i$ GIVEN BY TRAINING POINTS



$K(x, x_i) = e^{-\frac{\|x - x_i\|^2}{\sigma}}$

(3) INTERPRETATION OF KERNEL.

- IS THE INNER PRODUCT IN FEATURE SPACE
- MODELS A "DISTANCE" BETWEEN TWO INPUT POINTS x_i AND x_j.

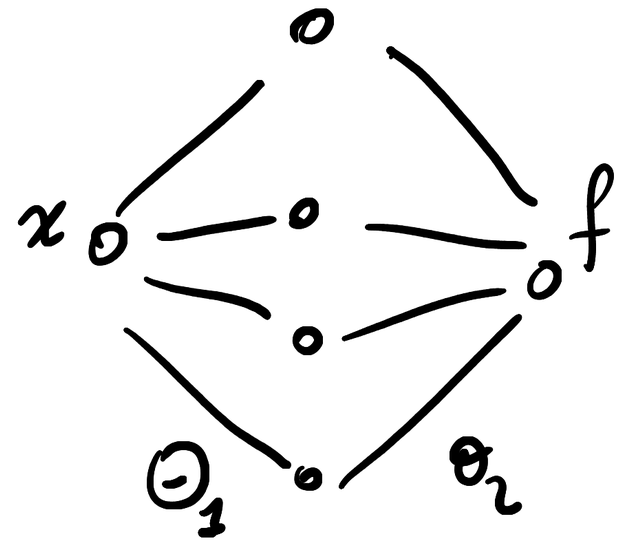
Neural Networks

Idea: In many problems, choosing the right feature map, kernel or knot points is hard.

↳ Why not learn them from data?

One hidden layer network:

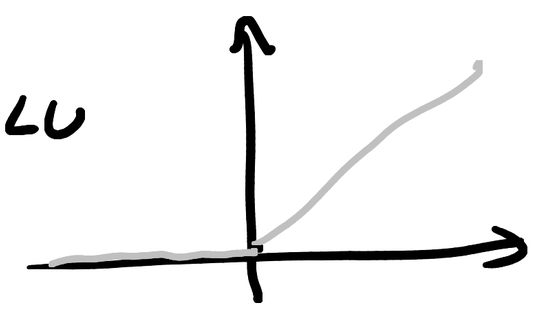
$$f_\theta(x) = \theta_2^T \sigma(\Theta_1 x + \theta_1)$$



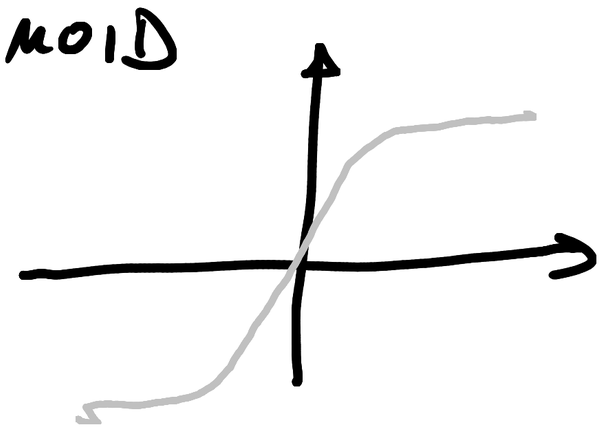
Feature mapping with learnable parameters $\Theta_1 \in \mathbb{R}^{p \times m}$ and $\theta_1 \in \mathbb{R}^m$

where σ is a non-linearity

- $\sigma(x) = \max(x, 0)$ ReLU



- $\sigma(x) = \frac{1}{1+e^{-x}}$ sigmoid



- UNIVERSAL APPROXIMATION THEOREM: WITH SUFFICIENT NEURONS p, $f_\theta(x)$ CAN APPROXIMATE ANY CONTINUOUS FUNCTION.

- IF WE FIX THE FIRST LAYER, THE NN BECOMES A SIMPLE FEATURE MAP.

- THERE IS NO CLOSED FORM SOLUTION TO THE LEARNING problem

$$\text{argmin} \sum_{i=1}^{N} (f_\theta(x_i) - y_i)^2$$

EVEN WORSE, THERE ARE MULTIPLE (POSSIBLY LOCAL) MINIMA.

$\Rightarrow$ WE GENERALLY USE GRADIENT DESCENT TO optimize THEM.

Deep Neural Networks ("AKA" Deep Learning):

- IDEA CONCATENATE MANY LAYERS

$$f_\theta(x) = \theta_L^T \sigma(\theta_{L-1} \cdots \sigma(\theta_1 x + \theta_1) + \theta_{L-1})$$

(L TIMES)

LEARNABLE PARAMETERS

$$\theta = \{\theta_1, \ldots, \theta_L, \theta_1, \ldots, \theta_{L-1}\}$$

- MOST NEURAL NETWORKS USED IN PRACTICE DO NOT TAKE THIS SIMPLE FORM.
  - ↳ CNNs (CONVOLUTIONAL NEURAL NETWORKS)
  - ↳ GNNs (GRAPH NEURAL NETWORKS)
  - ↳ TRANSFORMERS (ATTENTION-BASED NETWORKS)

- THIS ARCHITECTURES CAN BE (PARTLY) UNDERSTOOD THROUGH THE LENS OF <u>EQUIVARIANCE</u>.

SOME MATHEMATICAL CONCEPTS:

<u>Group Action</u>: THE ACTION OF A GROUP G IS THE SET OF INVERTIBLE TRANSFORMATIONS $\{T_g\}_{g \in G}$

WE MUST HAVE

① IDENTITY $T_e = Id$

② INVERSE $T_g^{-1} = T_{g^{-1}}$

③ COMPOSITION $T_{g_1} T_{g_2} = T_{g_1 \cdot g_2}$

EXAMPLES:

IMAGES
- TRANSLATIONS (IN 1D, 2D, ...)
- ROTATIONS (IN 2D, 3D)
- REFLECTIONS

GRAPHS
- PERMUTATIONS

EQUIVARIANCE: $f_\theta(x): \mathbb{R}^{n_{IN}} \to \mathbb{R}^{n_{OUT}}$ IS EQUIVARIANT TO $\{T_g\}_{g\in G}$ IF

$$f_\theta(T_g x) = \tilde{T}_g f_\theta(x)$$

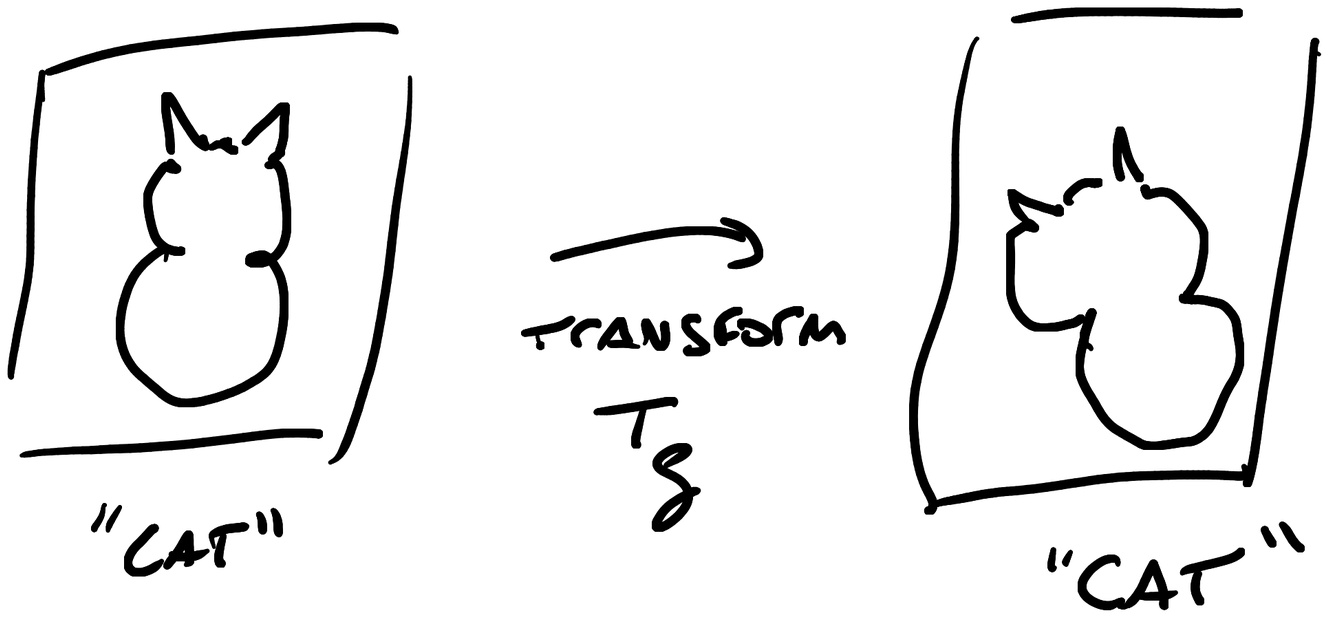
↳ VALID GROUP ACTION

INVARIANCE: A PARTICULAR CASE OF EQUIVARIANCE WHERE $\tilde{T}_g = Id \ \forall g \in G$ (TRIVIAL GROUP ACTION)

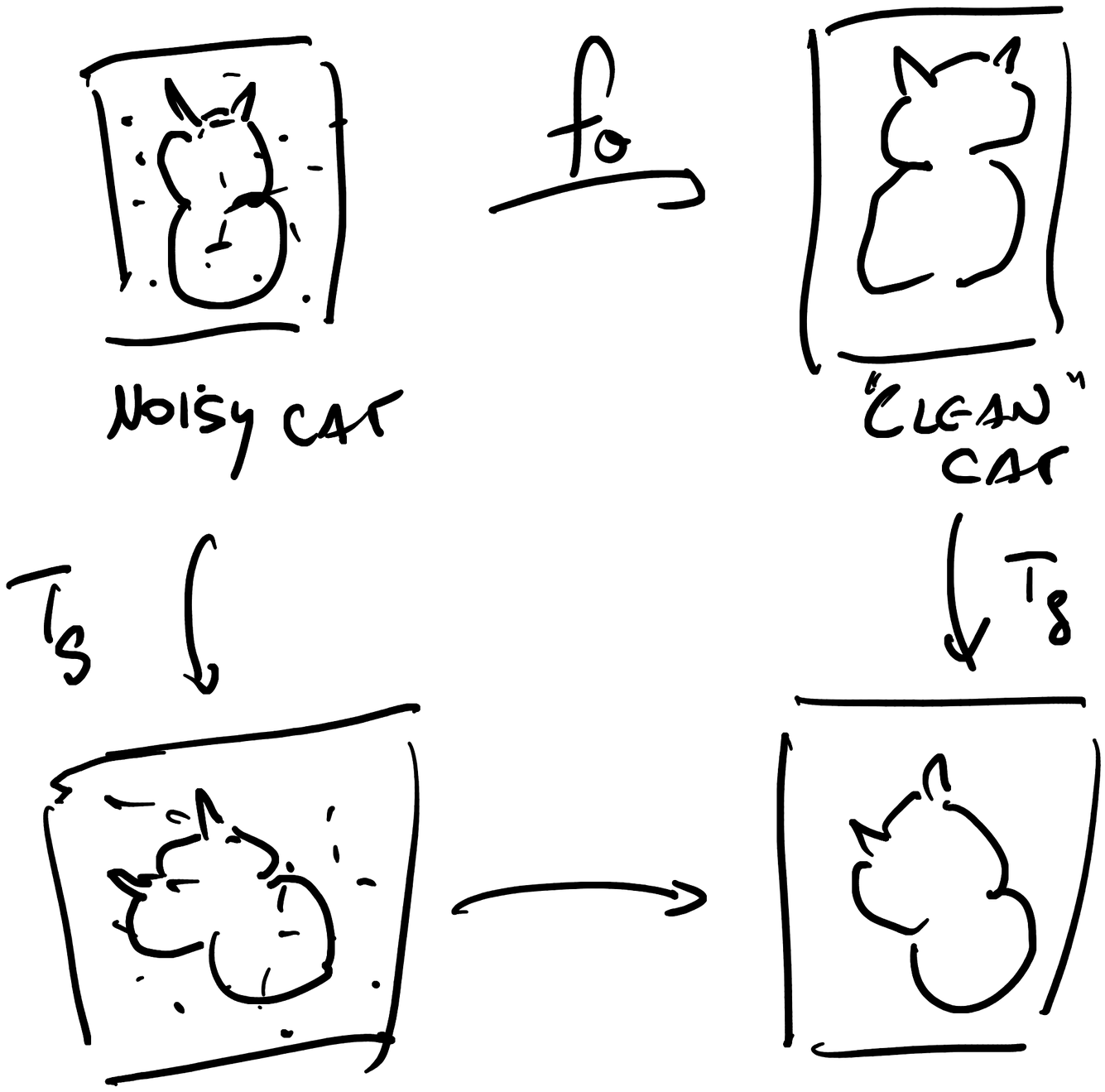
OR IN OTHER WORDS

$$f_\theta(T_g x) = f_\theta(x)$$

IN CLASSIFICATION TASKS, WE WANT AN INVARIANT FUNCTION $f_\theta(T_g x) = x$



IN REGRESSION TASKS, WE WANT AN EQUIVARIANT FUNCTION:



IDEA: PARAMETERIZE f_θ SO THAT IT IS INVARIANT/EQUIVARIANT FOR ANY POSSIBLE VALUE OF θ

(i.e. EVEN WITHOUT ANY TRAINING)

- THE COMPOSITION OF TWO EQUIVARIANT FUNCTIONS IS EQUIVARIANT.

MATHEMATICALLY $f = f_1 \circ f_2$

if f_1 AND f_2 ARE EQUIV. $\Rightarrow$ f is EQUIV.

A NEURAL NEWORK HAS 3 OPERATIONS

- BIAS: $f(x) = x + \theta$ WHICH IS ALWAYS EQUIVARIANT. ✓
- NON-LINEARITY: $f(x) = \sigma(x)$ WHICH IS EQUIVARIANT TO MOST ACTIONS IF APPLIED PIXEL-WISE

$$\sigma(x) = \begin{bmatrix} \sigma(x_1) \\ \vdots \\ \sigma(x_m) \end{bmatrix}$$

- LINEAR TRANSFORMATION

$$f(x) = \Theta x$$

WHICH IS NOT NECESSARILY EQUIVARIANT

FOR ALL $\Theta \in \mathbb{R}^{p_1 \times p_2}$

— WE CAN LOOK FOR MATRICES THAT VERIFY

$$S = \left\{ \Theta \in \mathbb{R}^{p_1 \times p_2} : \Theta T_g = \tilde{T}_g \Theta \quad \forall g \in G \right\}$$

THEOREM: S is A LINEAR SUBSPACE OF $\mathbb{R}^{p_1 \times p_2}$ AND THUS EQUIV. Θ CAN BE PARAMETERIZED AS

$$\Theta^{\text{equiv}} = \sum_{i=1}^{K} \theta_i \psi_i$$

WHERE $\psi_i \in \mathbb{R}^{p_1 \times p_2}$ ARE SOME BASIS

PARAMETER EFFICIENCY: $\frac{K}{p_1 p_2} < 1$

$K \rightarrow$ EQUIV. MATRICES

$p_1 p_2 \rightarrow$ ANY MATRIX.

If T_g are TRANSLATIONS, we get

$$\Theta \overset{\text{equiv.}}{=} \text{circ}(\theta)$$ CIRCULANT MATRIX

$$\text{circ}(\theta)\, x = \theta * x$$ CONVOLUTION

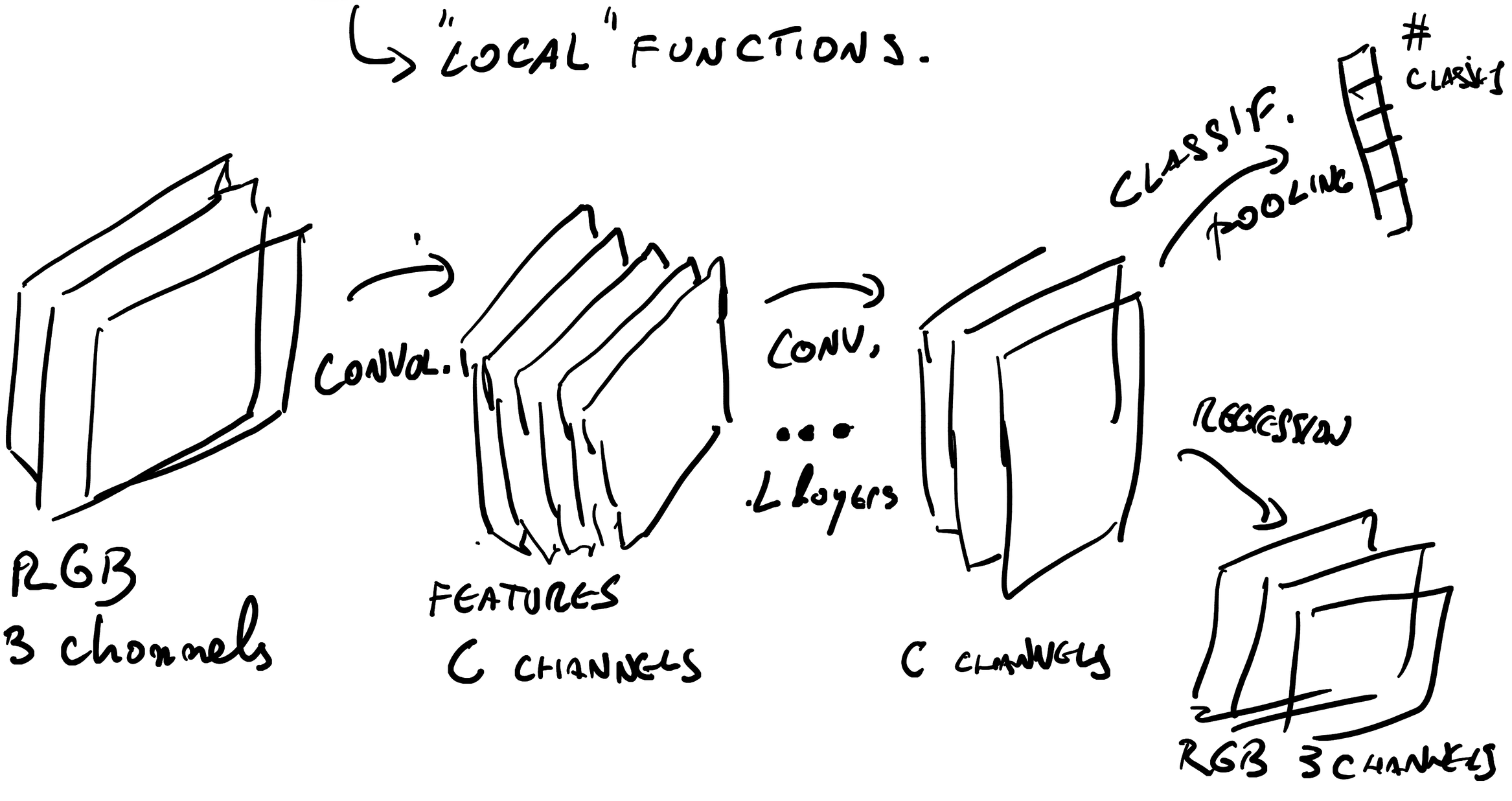
$\longrightarrow$ ONLY m PARAMETERS INSTEAD OF m^2

CNN

$$f_\theta(x) = \theta_L^T \sigma(\underbrace{\theta_{L-1} * \cdots \sigma(\theta_1 * x + \tilde{\theta}_1) + \tilde{\theta}_{L-1}}_{\text{EQUIVARIANT STEPS}})$$

- PARAMETERS θ_ℓ ARE CALLED FILTERS/WEIGHTS.
- WE GENERALLY CHOOSE THEM OF SMALL SIZE

e.g. $\underbrace{3 \times 3}_{\hookrightarrow \text{"LOCAL" FUNCTIONS.}}$ FOR IMAGES INSTEAD OF m PIXELS.

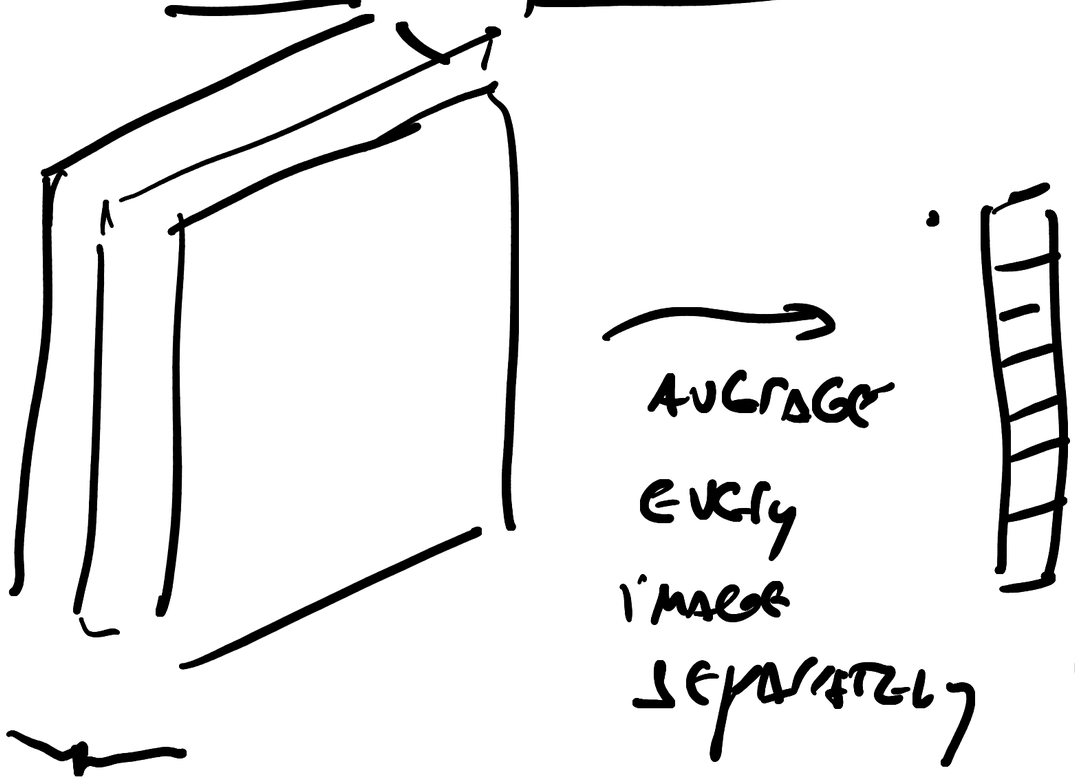


POOLING AVERAGE ACROSS THE IMAGE

FROM EQUIVARIANT → INVARIANT FUNCTIONS

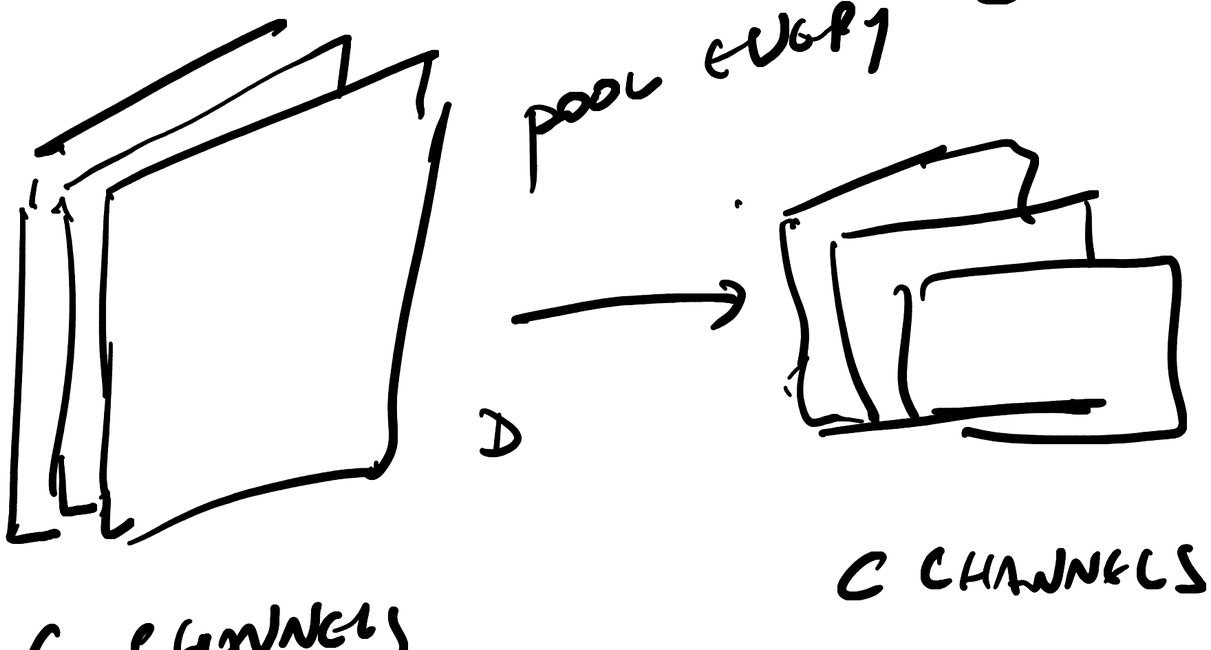
↳ TAKE AN AVERAGE OF THE WHOLE IMAGE

GLOBAL POOLING:



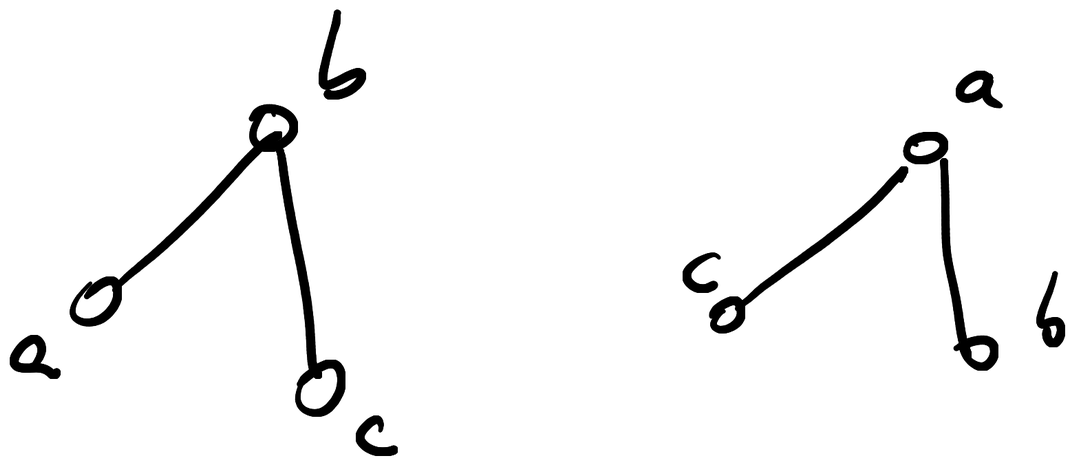
H × W PIXELS

LOCAL POOLING



INVARIANCE TO PERMUTATIONS

GRAPHS



ADJACENCY MATRIX

$$X_1 = \begin{array}{c} \\ a \\ b \\ c \end{array}\begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \end{array} \qquad X_2 = \begin{array}{c} \\ a \\ b \\ c \end{array}\begin{array}{c} \\ \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \end{array}$$

PERMUTATION

$$X_1 = T_g X_2 T_g^T$$

PERMUTATION MATRIX.

GRAPH NEUCAL NETWORKS

LINEAR LAYERS THAT ARE PERMUTATION INVARIANT.

UNSUPERVISED LEARNING

DATASET $\{x_i\}_{i=1:N}$ → NO LABEL DATA!

GOAL: LEARN SOMETHING ABOUT THE DATA GENERATION process $p(x)$.

1) LEARN THE SUPPORT $\mathcal{X} = \text{supp}\, p(x)$

↳ REAL-WORLD DATA IS INTEGRITY LOW-DIMENSIONAL ("MANIFOLD HYPOTHESIS")

2) LEARN A GENERATIVE MODEL

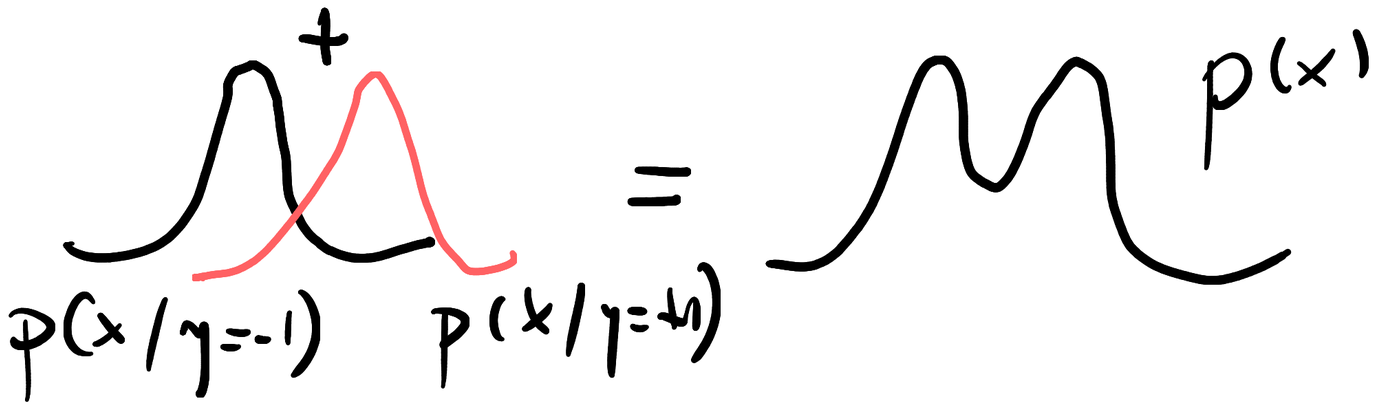
$$z \sim \mathcal{N}(0, I)$$

$$f_\theta(z) = x$$

f_θ: NETWORK THAT MAPS GAUSSIAN NOISE TO IMAGES.

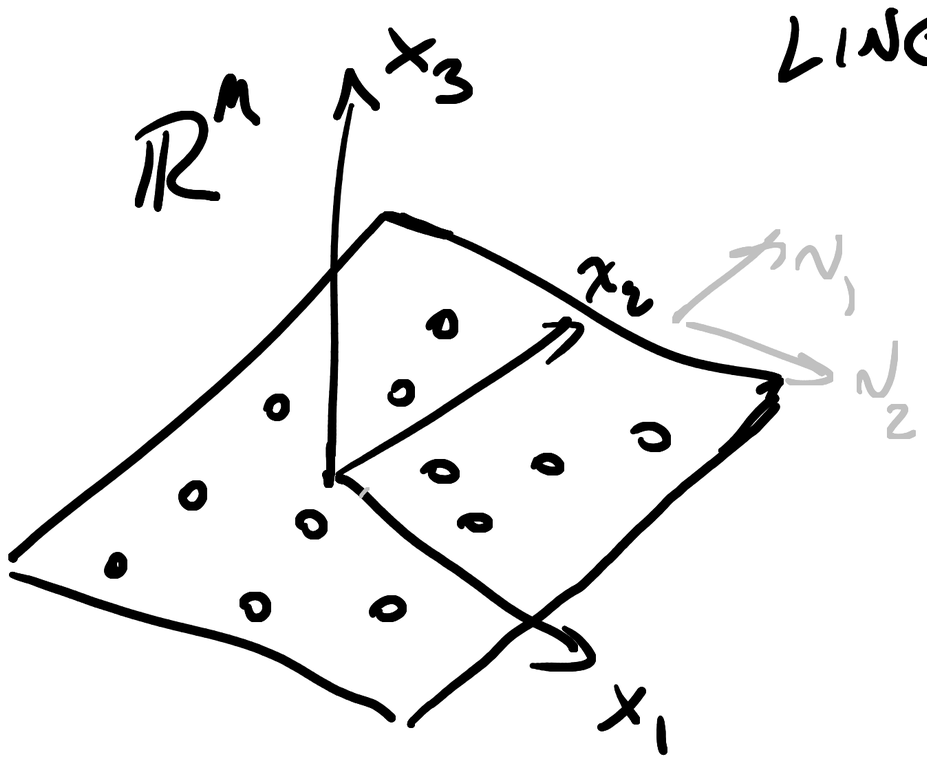
3) LEARN A CLUSTER MODEL

$p(x) \triangleq$ IS A MIXTURE OF DIFFERENT CLASSES



1) LEARNING THE SUPPORT $\mathcal{X}$ $(x \in \mathbb{R}^m)$

a) ASSUME $\mathcal{X}$ IS A K DIMENSIONAL LINEAR SUBSPACE



$$x_i = \underbrace{D}_{\text{BASIS OF } \mathcal{X}} \underbrace{V_i}_{\text{"COEFFICIENTS}} \qquad m \times K$$

DATA MATRIX

$$X = [x_1, \dots, x_N] \in \mathbb{R}^{m \times N}$$

$$X \overset{\text{SVD}}{=} D \cdot \operatorname{diag} S \, V^T \qquad D: m \times K \qquad S: K \times 1 \qquad V^T: K \times N$$

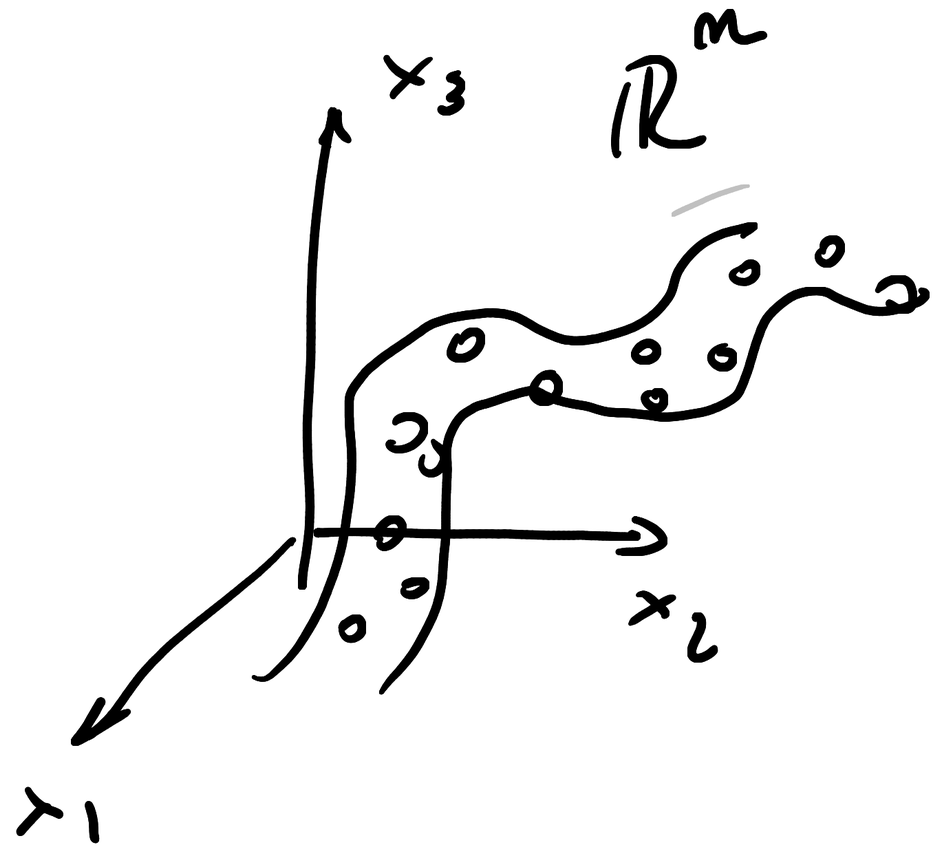
SINGULAR VALUE DECOMPOSITION

"PRINCIPAL COMPONENTS ANALYSIS.

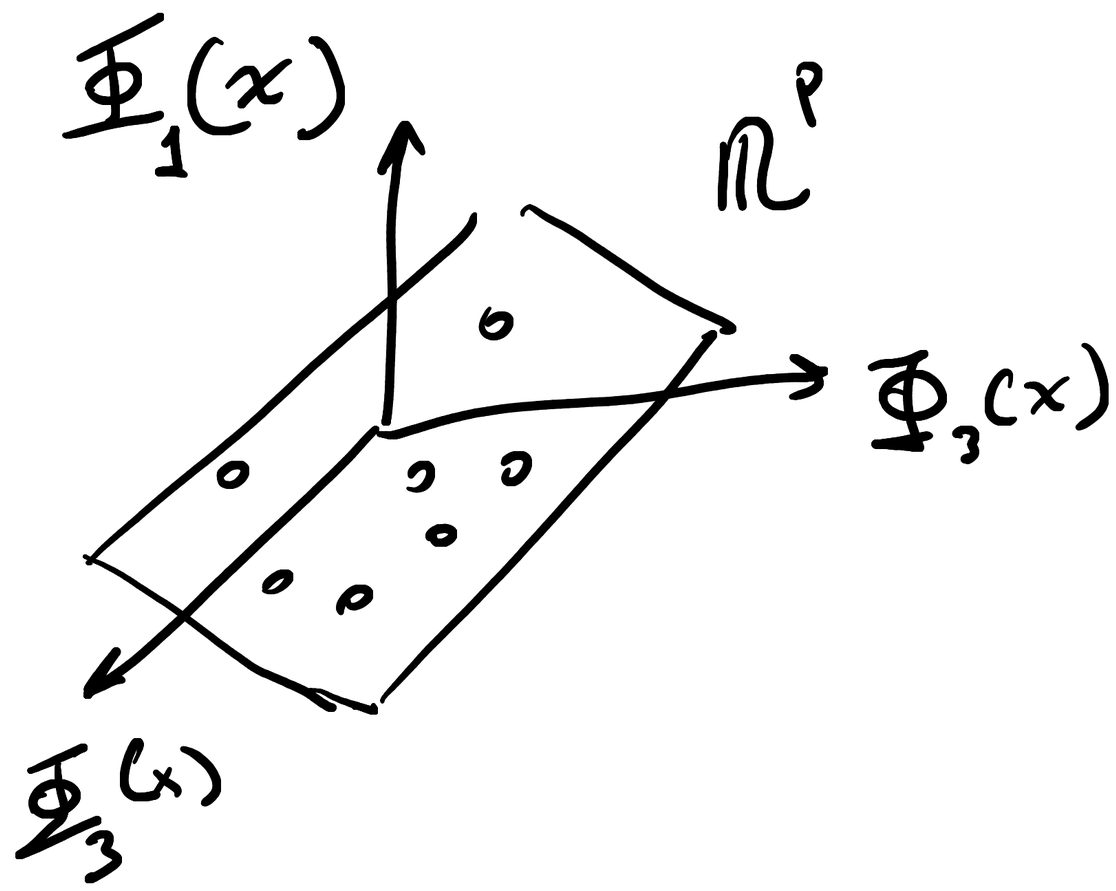
COEFFICIENTS ARE OBTAINED AS

$$V = \begin{bmatrix} \tilde{N}_1 \\ \tilde{N}_N \end{bmatrix} \Rightarrow N_i = \operatorname{diag} S \cdot \tilde{N}_i$$

b) $\mathcal{X}$ IS A K-DIMENSIONAL MANIFOLD



IDEA: MAP INTO FEATURE SPACE TO LINEARIZE WITH $\Phi: \mathbb{R}^m \to \mathbb{R}^p$



FEATURE MATRIX:

$$\Phi = [\Phi(x_1), \dots, \Phi(x_N)] \in \mathbb{R}^{p \times N}$$

$$\Phi \overset{\sim}{=} U \operatorname{diag} s \, V^T$$

↑ SINGULAR VALUE DECOMPOSITION

KERNEL PRINCIPAL COMPONENT ANALYSIS:

IDEA: USE THE KERNEL TRICK TO AVOID COMPUTING EXPLICIT FEATURE MAPS.

$$K = \Phi^T \Phi$$

↑ GRAM MATRIX $\in \mathbb{R}^{N \times N}$ (SYMMETRIC)

$$[K]_{ij} = k(x_i, x_j) \quad \text{WITH KERNEL } k.$$

EIGENVALUE DECOMPOSITION:

$$K = V \, d\log s^2 \, V^T$$

V: $m \times k$; $d\log s^2$: $k \times 1$; V^T: $k \times m$ → COEFFICIENTS IN NON-LINEAR BASE

(ASSUMING DATA IS CENTERED IN FEATURE SPACE)

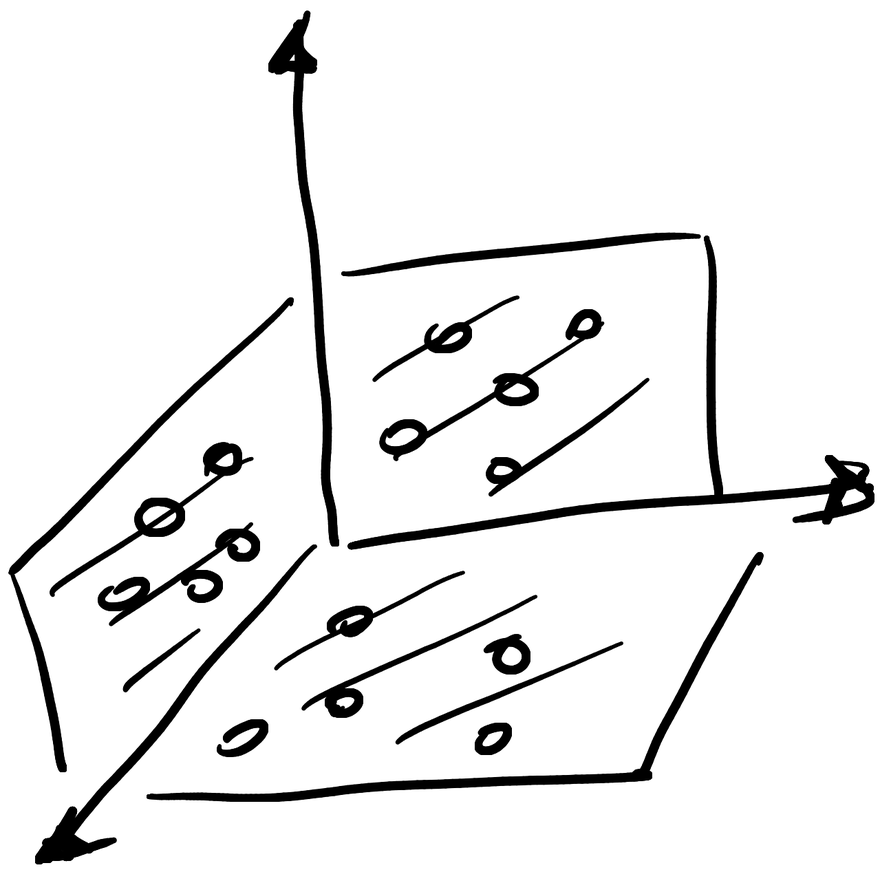
THIS DEFINES A MAPPING

$$x_i \in \mathbb{R}^m \xrightarrow{f} N_i \in \mathbb{R}^k$$

IF DATA IS NOT CENTERED IN FEATURE SPACE WE USE

$$K' = K - \underbrace{1_{N\times N}\frac{K}{N} - \frac{K}{N}1_{N\times N} + 1_{N\times N}\frac{K}{N^2}1_{N\times N}}_{\text{CENTERING.}}$$

c) ASSUME $\mathcal{X}$ IS A UNION-OF-SUBSPACES OF DIMENSION K (UoS)



IN PARTICULAR, A POPULAR UoS MODEL IS THE SPARSE DICTIONARY MODEL

$$x = Dz$$

WHERE $D \in \mathbb{R}^{n \times p}$ IS A DICTIONARY

AND z IS K-SPARSE

$(\|z\|_0 \leq K)$

TYPICAL DICTIONARY LEARNING ALGORITHMS WORK IN 2 STEPS:

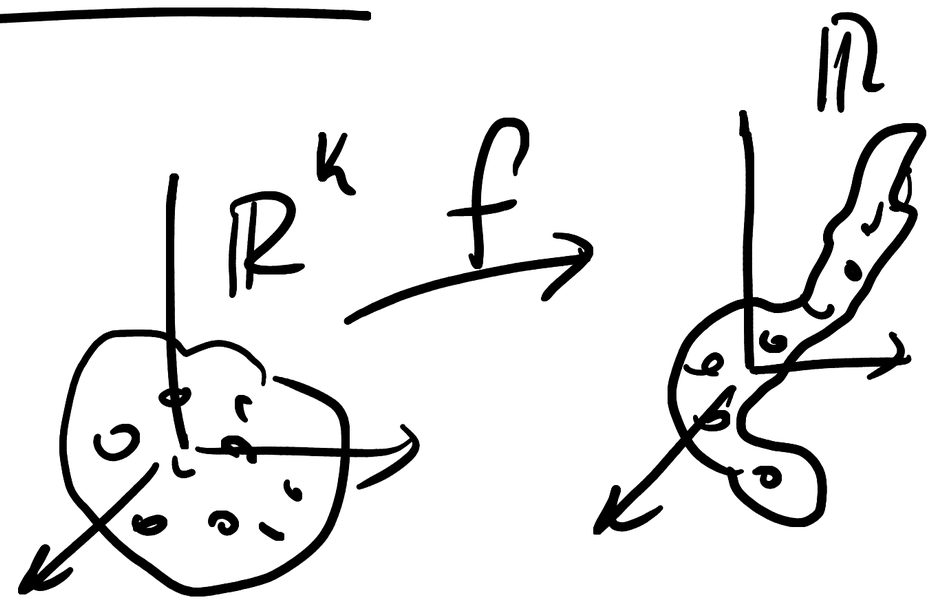
1) SPARSE CODING WITH D FIX $\rightarrow$ FIND $Z \in \mathbb{R}^{m \times N}$ WITH SPARSE COLUMNS.

for example $\hat{Z} = \underset{Z}{\operatorname{argmin}} \; \|X - DZ\|^2 + \lambda \|Z\|_1$

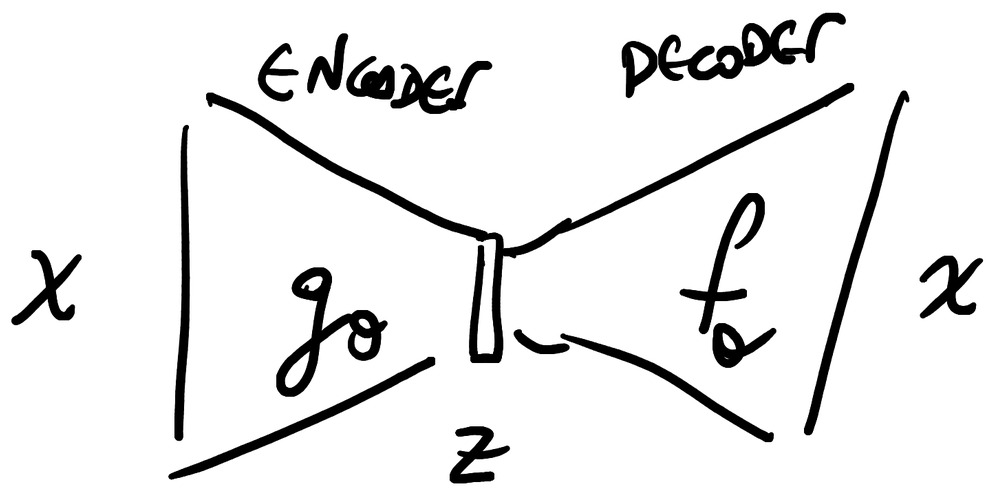
2) DICTIONARY UPDATE WITH Z FIX

$$D = \underset{D}{\operatorname{argmin}} \; \|X - DZ\|^2$$

GENERATIVE MODELS:



1) AUTOENCODERS

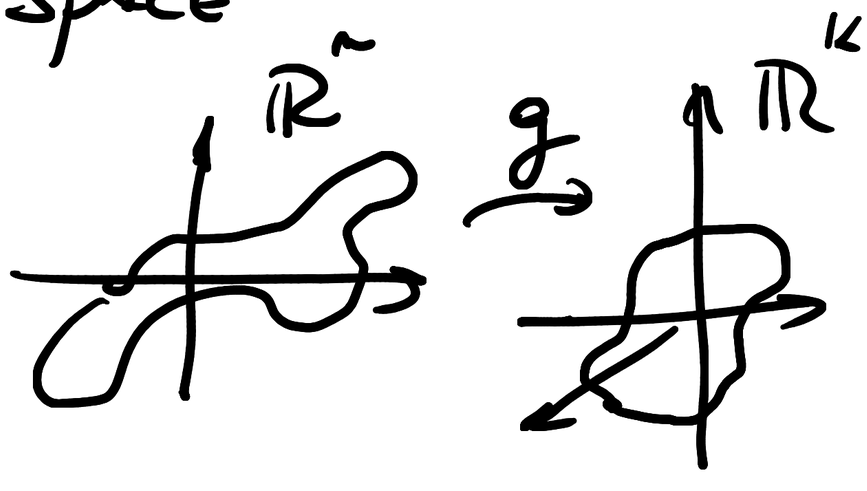


$$\underset{f,g}{\operatorname{argmin}} \; \|f_\phi \circ g_\theta(x) - x\|^2$$

generate data $z \sim N(0, I)$

$\tilde{x} = f_\phi(z)$

THIS APPROACH FAILS if THE DATA IN THE ENCODED SPACE (LATENT) IS NOT GAUSSIAN.



VARIATIONAL AUTOENCODERS

- TRAIN THE AUTOENCODER TO GENERATE A GAUSSIAN LATENT SPACE.
- PROBABILISTIC FORMULATION

$$x/z \sim \mathcal{N}\left(f_\theta(z),\ h_\theta(z)\, I\right)$$

$h_\theta(z)$ ↑ VARIANCE NETWORK

TRAINING OBJECTIVE

$$\arg\min_\theta \sum_{i=1}^{n} \| f_\theta \circ g_\phi(x_i) - x_i \|^2 + KL\left(\mathcal{N}(f_\theta, h_\theta^2),\ \mathcal{N}(0, I)\right)$$

KULLBACK LEIBLER DIVERGENCE ("DISTANCE" BETWEEN TWO DISTRIBUTIONS)

$$KL\left(N(f_\theta, h_\theta^2), N(0, I)\right)$$

$$\propto \sum_{i=1} \| f_\theta(x_i) \|^2 - \frac{n}{2} h_\theta^2(x_i) - 2n \log h_\theta(x_i)$$

ADDITIONAL REGULARIZATION TERM TO OBTAIN A "GAUSSIAN" LATENT SPACE.

GENERATIVE ADVERSARIAL NETWORKS (GANs)

WHY CAN'T WE JUST LEARN THE DISTRIBUTION AS:

$$\underset{f}{\text{arg.min}} \; KL\left(\underbrace{f_{\theta\#} N(0, I)}_{\text{DISTRIBUTION}}, \underbrace{p(x)}_{\substack{\text{APPROXIMATED} \\ \text{BY } \sum_{i=1}^{N} \frac{\delta_{x_i}}{N} \\ \text{(EMPIRICAL MEASURE)}}} \right)$$

$$\begin{cases} z \sim N(0, I) \\ x = f_\theta(z) \end{cases}$$

$\Rightarrow$ KL IS NOT A GOOD "DISTANCE"

↳ REPLACE KL BY A LEARNED DISTANCE d_ϕ

$d_\theta : \mathbb{R}^m \to \mathbb{R}$ IS THE DISCRIMINATOR NETWORK WHICH SCORES IF AN IMAGE x BELONGS TO $p(x)$ OR IF IT IS FAKE

$$\min_{f_\theta} \max_{d_\theta} \underbrace{\mathbb{E}_{x \sim p(x)} \log d_\theta(x)}_{\text{TRAINS THE DISCRIMINATOR TO PREFER REAL DATA}} + \underbrace{\mathbb{E}_{z \sim N(0,1)} \log(1 - d_\theta(f_\theta(z)))}_{\text{TRAINS DISCRIMINATOR TO "FOOL" THE DISCRIMINATOR}}$$

EXAMPLES OF GANS: STYLEGAN 2, BIGGAN, ETC.

DIFFUSION MODELS:

IDEA: A DENOISER NETWORK $f_\theta : \mathbb{R}^n \to \mathbb{R}^n$ (implicitly) STORES INFORMATION ABOUT $p(x)$.

↳ STARTING WITH NOISE $z^1 \sim N(0, I)$ AND $x^1 = 0$

$$\begin{cases} z^k \sim N(0, I\sigma_k^2) \\ x^{k+1} = f_\theta(x^k + z^k) \end{cases}$$

MATHEMATICAL INTUITION (TWEEDIE'S FORMULA):

A MINIMUM MEAN SQUARED ERROR DENOISER, i.e.

$$f_\theta \approx \arg\min_\theta \mathbb{E}_{x\varepsilon}\|f(x+\varepsilon) - x\|^2$$

IS RELATED TO A GRADIENT ASCENT ON $p(x)$:

$$f(x) = x + \sigma \nabla_x \log p_\sigma(x)$$

WHERE $p_\sigma(x)$ IS THE DISTRIBUTION of $x + \sigma\varepsilon$, $\varepsilon \sim N(0, I)$

- By applying multiple times f_θ, we are increasing the likelihood that x belongs to $p(x)$.