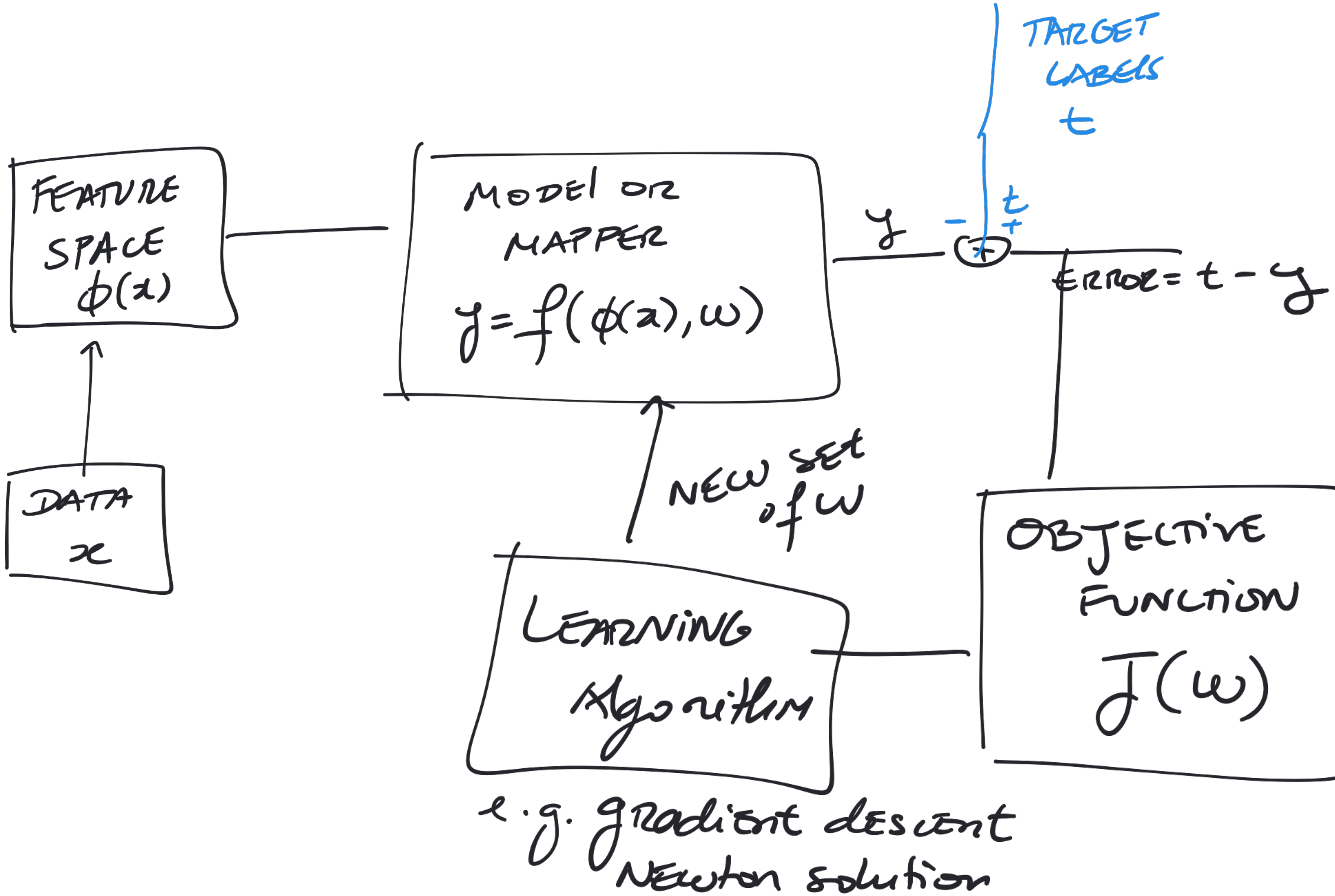

FEATURE SPACE
$\phi(x)$
DATA
x
MODEL OR MAPPER
$y = f(\phi(x), w)$
y
TARGET LABELS
t
$-$
t
$+$
ERROR $= t - y$
NEW SET OF w
LEARNING Algorithm
OBJECTIVE FUNCTION
$J(w)$
e.g. gradient descent
Newton solution

Linear Regression w/ Polynomial Features

↳ supervised learning

TRAINING DATA : $\{(x_i, t_i)\}_{i=1}^{N}$, $x_i \in \mathbb{R}$
$t_i \in \mathbb{R}$

MODEL OR MAPPER : $y = f(\phi(x), w)$, $w \equiv$ parameters of the model

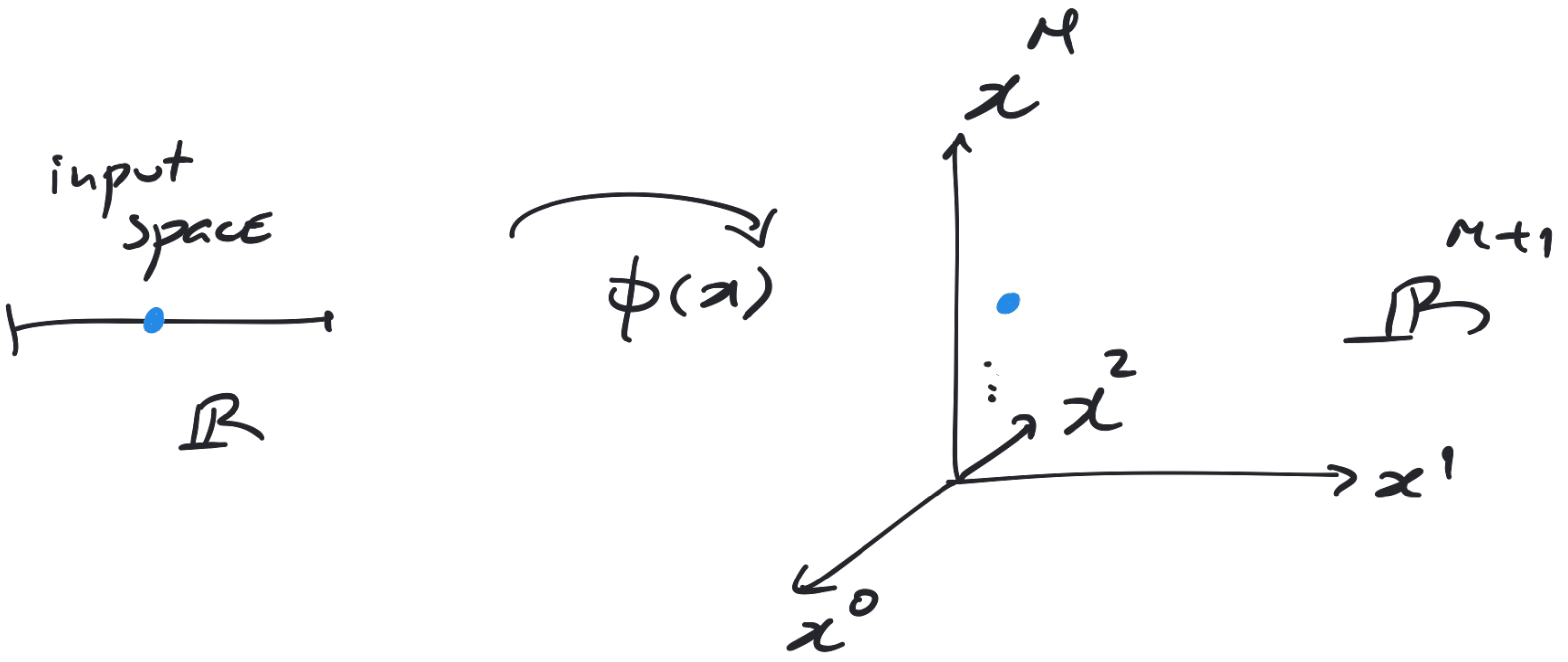
$$y = w_0 + w_1 \cdot x + w_2 \cdot x^2 = w^T \cdot \phi(x) = \phi^T(x) \cdot w$$

Polynomial Basis function : $\phi(x) = [x^0, x^1, x^2]^T$

PARAMETERS : $w = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}$

In general, the dimensionality of feature space is determined by the hyperparameter

$M \equiv$ model order



Hyperparameters $\leftarrow$ are user-defined

parameters $\leftarrow$ are found using the hyperparameters for a given training set.

For each training sample x_i:

$$\phi(x_i) = [x_i^0, x_i^1, x_i^2, \dots, x_i^M]^T$$

MAPPER: $\boxed{y(x_i) = \phi^T(x_i) \cdot w}$

$$\begin{cases} y(x_1) = \phi^T(x_1) \cdot w \\ y(x_2) = \phi^T(x_2) \cdot w \\ \vdots \\ y(x_N) = \phi^T(x_N) \cdot w \end{cases}$$

$N \equiv$ # samples in training

$M \equiv$ hyperparameter

FEATURE MATRIX : $X = \begin{bmatrix} \phi^T(x_1) \\ \phi^T(x_2) \\ \vdots \\ \phi^T(x_N) \end{bmatrix}_{N \times (M+1)}$

$$X = \begin{bmatrix} x_1^0 & x_1^1 & x_1^2 & \cdots & x_1^M \\ x_2^0 & x_2^0 & x_2^2 & \cdots & x_2^M \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_N^0 & x_N^1 & x_N^2 & \cdots & x_N^M \end{bmatrix}, \quad w = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_M \end{bmatrix}_{(M+1) \times 1}$$

$$\boxed{y = X \cdot w}$$

y is $N \times 1$

Let t be the vector with the desired / target values :

$$t = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{bmatrix}_{N \times 1}$$

Error of prediction : $\underset{N \times 1}{E} = t - y$

MEAN SQUARED ERROR

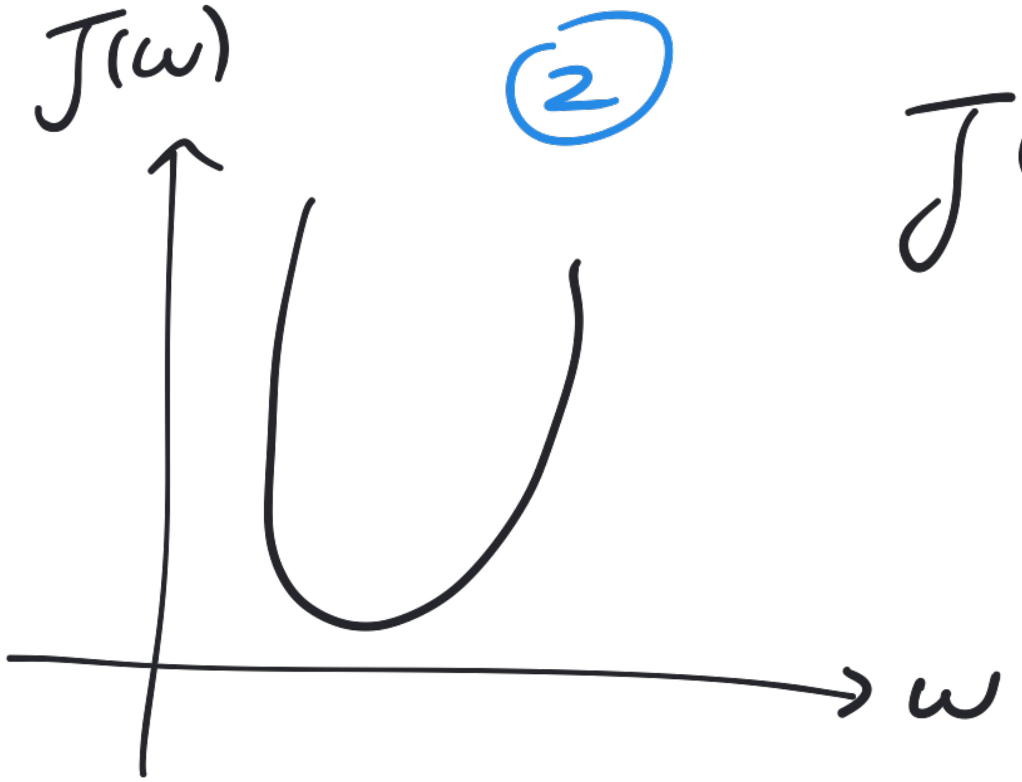
OBJECTIVE FUNCTION : 1

$$\frac{1}{N} \sum_{i=1}^{N} \varepsilon_i^2 = J(w)$$

MEAN ABSOLUTE ERROR (MAE)

2

$$J(w) = \frac{1}{N} \sum_{i=1}^{N} |\varepsilon_i|$$



Consider the objective function:

$$J(w) = \frac{1}{2} \sum_{i=1}^{N} \varepsilon_i^2$$

$$= \frac{1}{2} \sum_{i=1}^{N} (t_i - y_i)^2$$

$$= \frac{1}{2} \sum_{i=1}^{N} (t_i - \phi^T(x_i) . w)^2$$

$$= \frac{1}{2} (t - X w)^T (t - X w)$$

$$= \frac{1}{2} \| t - X w \|_2^2$$

Reminder:

$\|x\|_2^2 = x^T . x$

$\|x\|_2 = (x_1^2 + x_2^2 + \dots + x_N^2)^{1/2}$

$\|x\|_2 \equiv$ L2-norm

Learning Algorithm

Pose the question:

$$\arg\min_{w} J(w) = w^*$$

Necessary condition: $$\frac{\partial J(w^*)}{\partial w} = 0$$

$$J(w) = \frac{1}{2} \| t - Xw \|_2^2$$
$$= \frac{1}{2} (t - Xw)^T (t - Xw)$$
$$= \frac{1}{2} (t^T - w^T . X^T)(t - Xw)$$
$$= \frac{1}{2} (t^T . t - t^T . X w - w^T X^T . t + w^T X^T X w)$$

Reminder:

$(AB)^T = B^T . A^T$

$(AB^T)^T = B . A^T$

$\frac{d\, X.w}{dw} = X$, $\frac{d\, X w^T}{\partial w} = X^T$

$$\frac{\partial J(w)}{\partial w} = 0 \iff - t^T . X - (X^T . t)^T + (X^T X . w)^T + w^T X^T X = 0$$
$$\iff - t^T X - t^T . X + w^T X^T X + w^T X^T X = 0$$
$$\iff - t^T X + w^T X^T X = 0$$
$$\iff - X^T . t + X^T X . w = 0 \implies (*)$$

applying transposes

$$(*) \Leftrightarrow \quad X^T X w = X^T \cdot t$$

$$\Leftrightarrow \quad \boxed{w = \underbrace{(X^T X)^{-1} \cdot X^T}_{\text{pseudo-inverse of } X} \cdot t}$$

if $(X^T X)^{-1}$ exists

pseudo-inverse of X

denoted as $X^\dagger$ (X "dagger")

X is (typically) a tall matrix

$$\Leftrightarrow \quad \boxed{w = X^\dagger \cdot t}$$

Food for thought:

① In what scenarios is $X^T X$ not invertable?

② Can you force it to be invertible? How?