ML Curve Fitting Perspective of the World

Dr. Kalidas Y., IIT Tirupati

(C) Dr. Kalidas Y., IIT Tirupati

Find equation of line passing through two points

- Point A: (0,4)
- Point B: (1,7)

Find equation of line passing through two points

- Point A: (0,4)
- Point B: (1,7)
- Line is: $y = 3x+4$

Find equation of line passing through three points

- Point A: (0,4)
- Point B: (1,7)
- Point C: (-1,1)

Happy go lucky!

Find equation of line passing through three points..

- Point A: (0,4)
- Point B: (1,7)
- ~~Point C: (-1,1)~~ Point C: (-1,2)

~~**Happy go lucky!**~~

Through an error saying no line can pass through the given three points i.e. they are not collinear

Find equation of line passing through *just one*

- Point A: (0,4)
- ~~Point B: (1,7)~~
- ~~Point C: (-1,1)~~ ~~Point C: (-1,2)~~

~~**Happy go lucky!**~~

~~**Through an error saying no line can pass through the given three points i.e. they are not collinear**~~

Infinite number of solutions!

Non ML World?

RANK of the matrix X

$$X_{N\times k} W_{k\times 1} = y_{N\times 1}$$

There are k variables

There are N equations

Rank of X is r

When there is NO SOLUTION → r > k

When there are INFINITE NUMBER OF SOLUTIONS → r < k

(C) Dr. Kalidas Y., IIT Tirupati

ML Wold?

~~*RANK of the matrix X*~~

$$X_{N\times k} W_{k\times 1} = y_{N\times 1}$$

LOSS FUNCTION

There are k ~~***variables***~~ ***parameters***

There are N ~~**equations**~~ ***terms of the loss function***

~~**Rank of X is r**~~

~~**When there is NO SOLUTION → r > k**~~

~~**When there are INFINITE NUMBER OF SOLUTIONS → r < k**~~

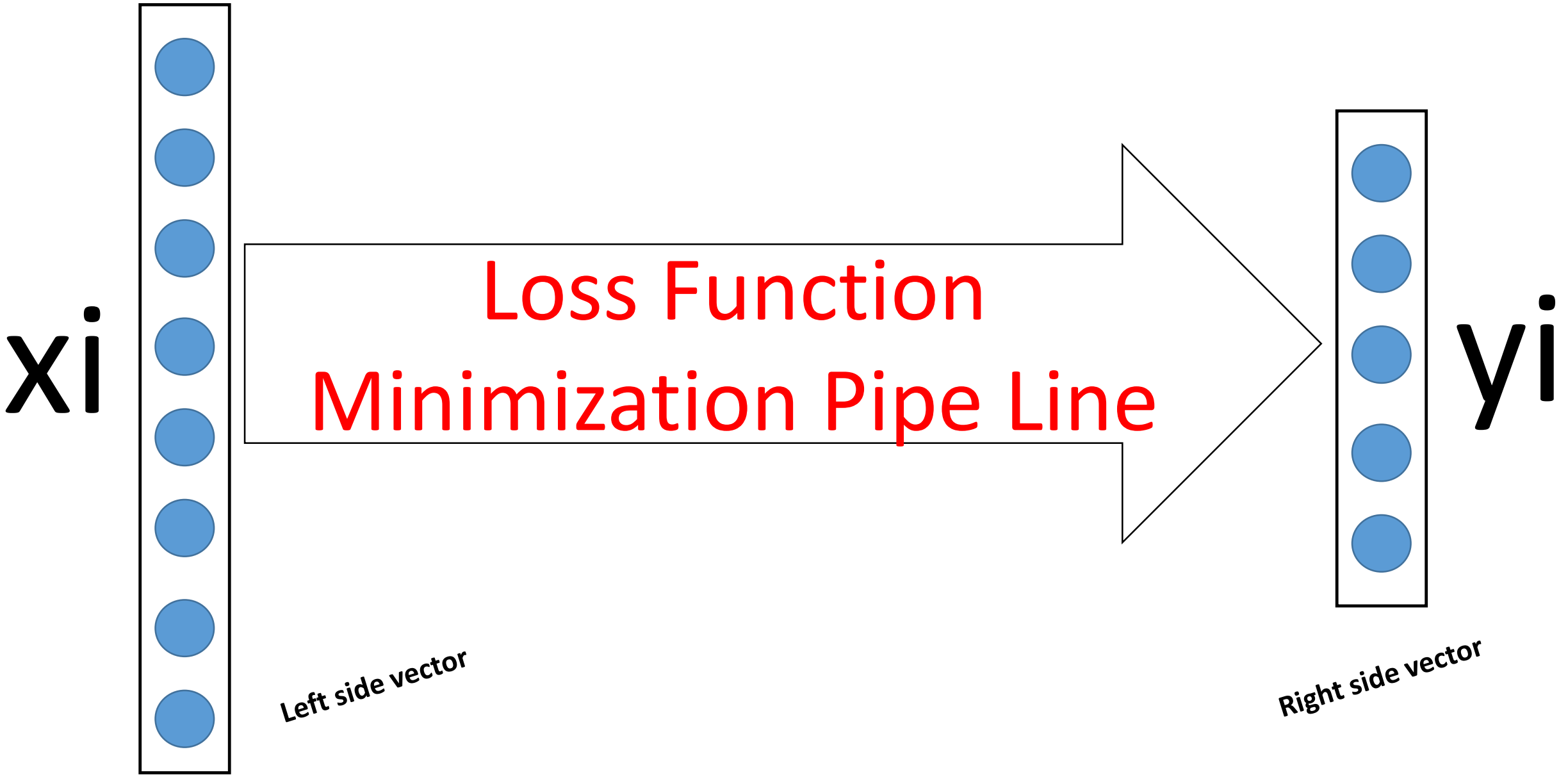
(C) Dr. Kalidas Y., IIT Tirupati

Loss function perspective → ML world

$$L(w) = (Xw - y)^T (Xw - y)$$

$$L(w) = \sum_{i=1}^{i=N} \left(y_i - \left(\sum_{j=1}^{j=k} w_j \, x_{i,j} \right) \right)^2$$

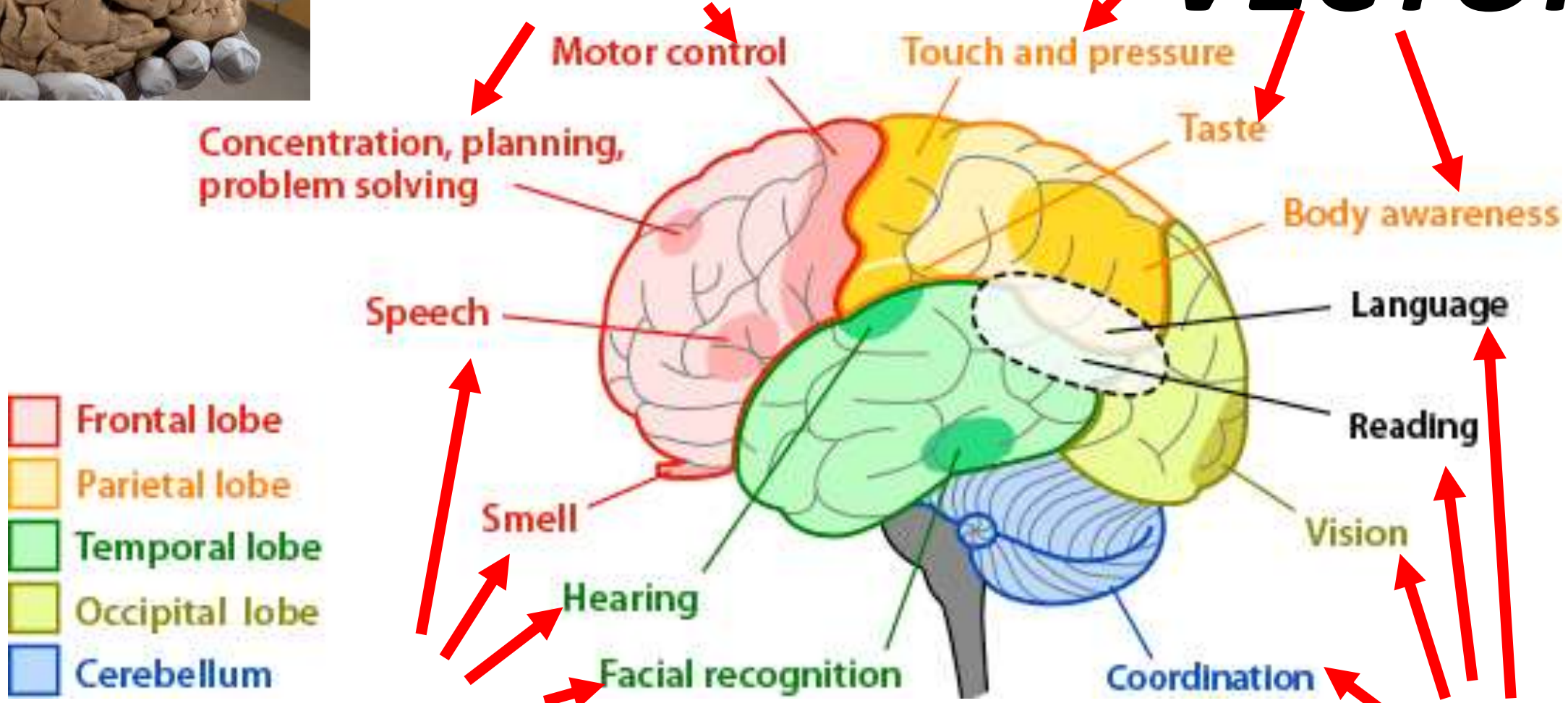
$$w_{(new)} = w_{(old)} - \eta \nabla L \Big|_{w=w_{(old)}}$$



(C) Dr. Kalidas Y., IIT Tirupati

VECTOR

VECTOR

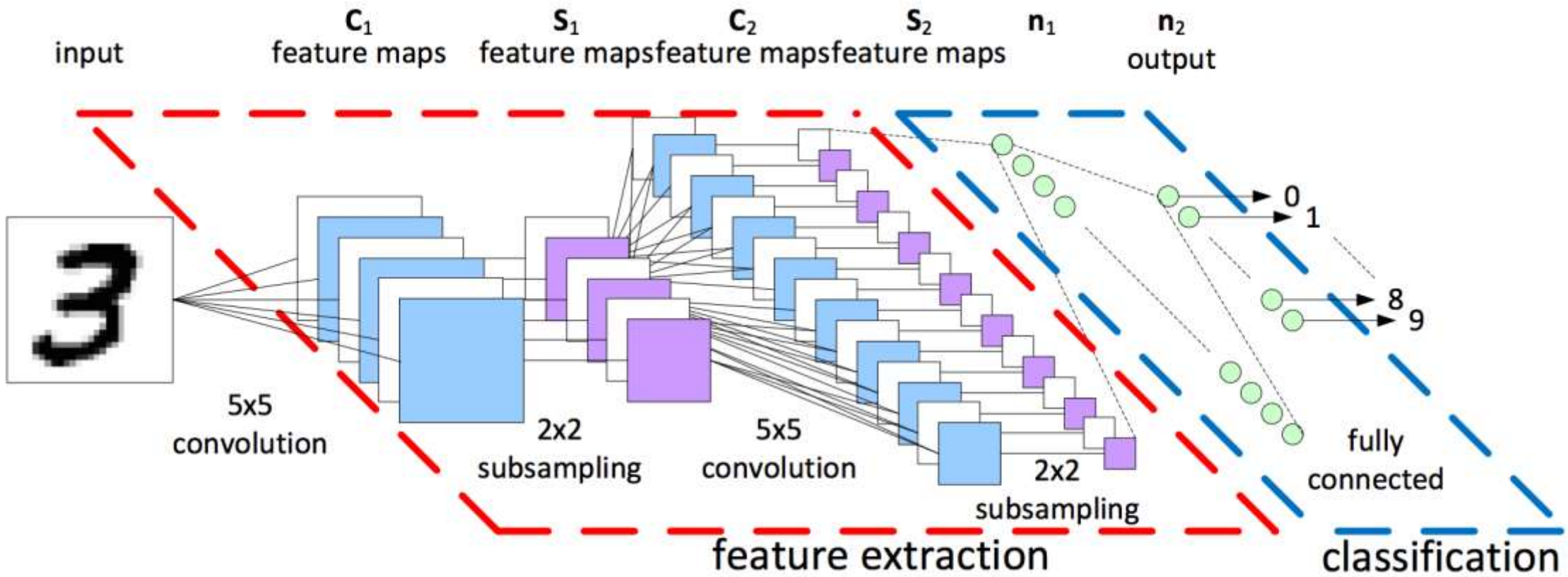


VECTOR

VECTOR

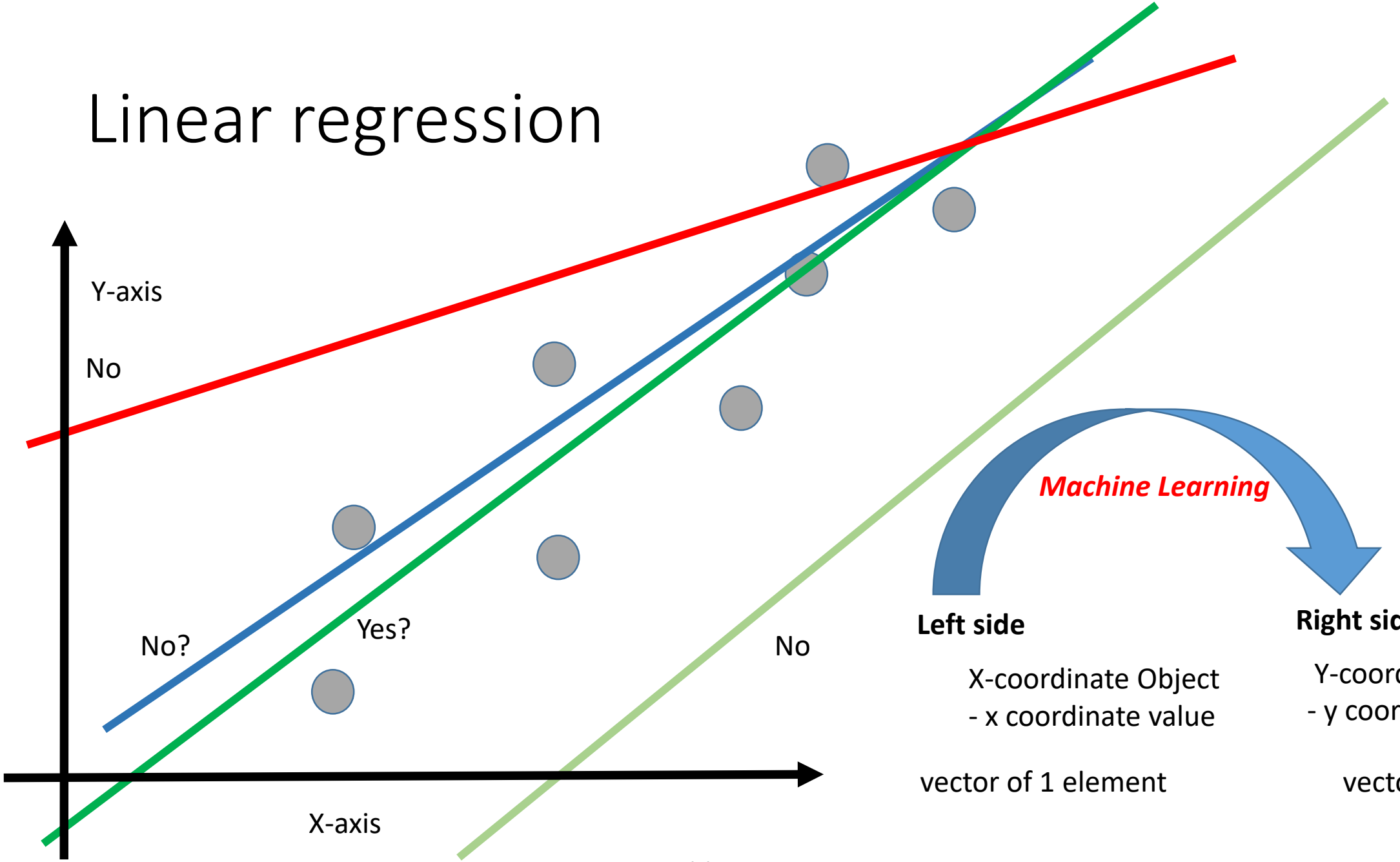
Reference – link here anyway and here.

(C) Dr. Kalidas Y., IIT Tirupati



Ref image here.

Linear regression



Machine Learning

Left side

X-coordinate Object
- x coordinate value

vector of 1 element

Right side

Y-coordinate Object
- y coordinate value

vector of 1 element

What are the different ways?

2nd argument is the *list of Parameters*

```
SteepestDescentSolver( L, [m1,c1], gradL)
```

```
//m1,c1 = some initial value is already input to the function
```

- **Iterate 1000 steps (*or whatever*)**
- **IF L(m1,c1) < Threshold**
  - **RETURN**
- **[m1,c1] = [m1,c1] – gradL(m1,c1)**

1st argument is the Loss function

3rd argument is the gradient function

This function outputs a list or vector

(C) Dr. Kalidas Y., IIT Tirupati.

Defining a loss function

```
# Author: Kalidas Y <ykalidas at iittp dot a
c dot in>
# License: BSD 3 clause

def fun_error(w) :
  diff = np.matmul(w, X.T) - y.T
  err_val = np.matmul( diff, diff.T )/X.shap
e[0]
  return err_val[0][0]
```

Defining Gradient Function

```
# Author: Kalidas Y <ykalidas at iittp dot ac dot
in>
# License: BSD 3 clause

def fun_grad_error(w) :
  grad_val = 2*np.matmul( np.matmul(w, X.T) -
 y.T, X )/X.shape[0]
  return np.concatenate(grad_val)
```

Invoking Optimization function

```
# Author: Kalidas Y <ykalidas at iittp dot ac dot
in>
# License: BSD 3 clause

from scipy.optimize import minimize

w_init = np.random.randn(1,2)

res = minimize(fun=fun_error, x0=w_init, jac=fun_g
rad_error,method='BFGS')
```

Analytical Solution… another numerical method…

which is not much useful in ML formulation

1. $Xw = y$
2. $X^T(Xw) = X^T(y)$
3. $(X^TX)w = X^Ty$
4. $(X^TX)^{-1}(X^TX)w = (X^TX)^{-1}X^Ty$
5. $w = (X^TX)^{-1}X^Ty$

There are numerical methods to compute inverse of a matrix

e.g. ***Schultz Method***

$$Z^{(t+1)} = 2\,\alpha\, Z^{(t)} - \beta\, Z^{(t)} X\, Z^{(t)}$$

$$Z^{(0)} = rand()$$

Formulation

```
In [ ]: # Author: Kalidas Y <ykalidas at iittp dot ac dot in>
        # License: BSD 3 clause

        w_analytical = np.matmul( np.matmul( np.linalg.inv( np.matmul (X.T, X) ), X.T ), y)

        print (w_analytical)
```

```
[[3.10219092]
```