Deep Learning (1470)

Randall Balestriero

Class 3

What can we do?

Featurization

What can we do?

Featurization

- Add all sort of nonlinear transforms of the data prior to your linear model

What can we do?

Featurization

- Add all sort of nonlinear transforms of the data prior to your linear model
- $\mathbf{x}_n \leftarrow (\mathbf{x}_n, \mathbf{x}_n^2, \log(\mathbf{x}_n), \ldots) \in \mathbb{R}^{D'}$

What can we do?

Featurization

- Add all sort of nonlinear transforms of the data prior to your linear model
- $\mathbf{x}_n \leftarrow (\mathbf{x}_n, \mathbf{x}_n^2, \log(\mathbf{x}_n), \ldots) \in \mathbb{R}^{D'}$
- If $\mathrm{rank}([\mathbf{x}_1, \ldots, \mathbf{x}_N]) \geq N$ then the training loss is 0!

What can we do?

Featurization

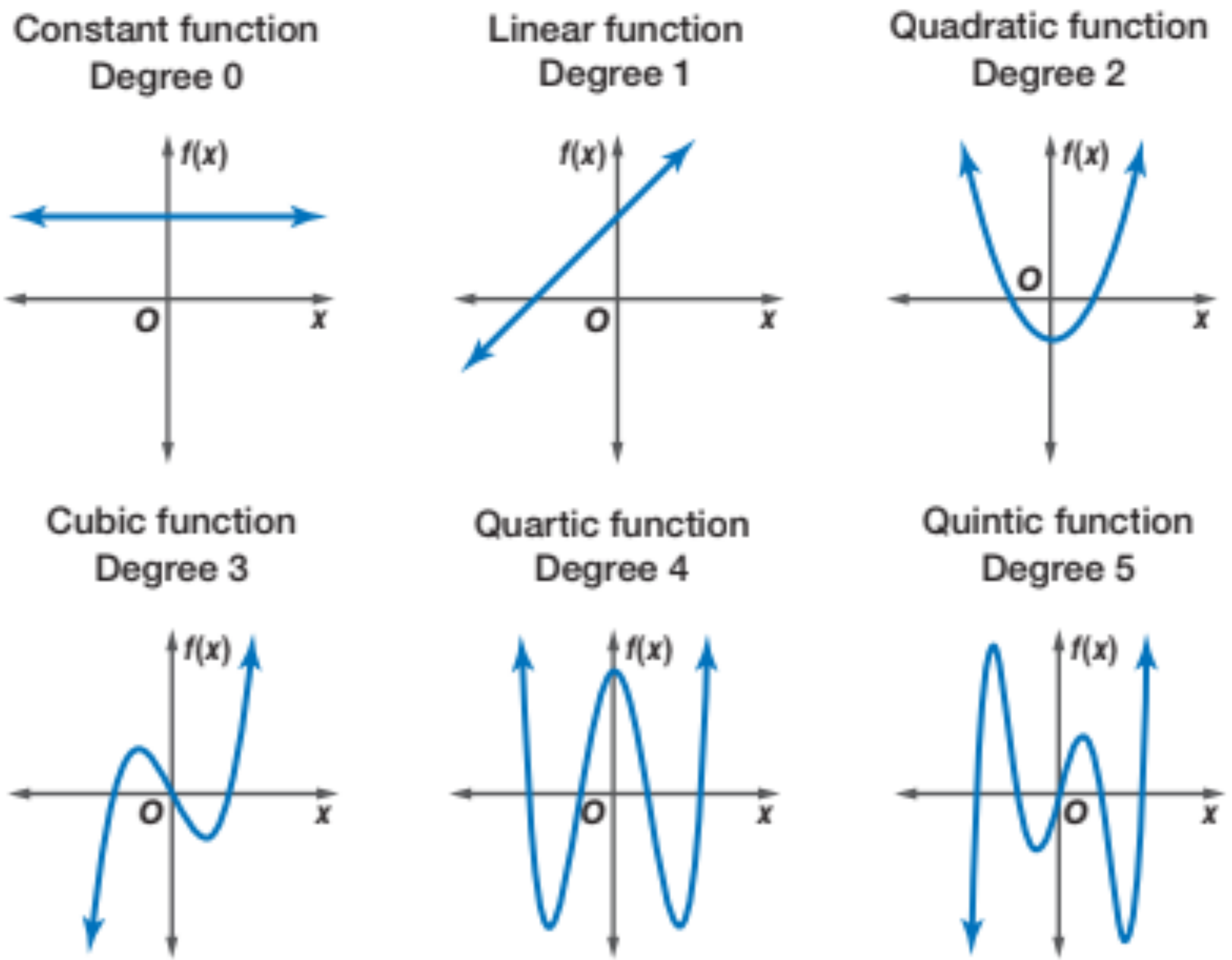
- Add all sort of nonlinear transforms of the data prior to your linear model
- $\mathbf{x}_n \leftarrow (\mathbf{x}_n, \mathbf{x}_n^2, \log(\mathbf{x}_n), \ldots) \in \mathbb{R}^{D'}$
- If $\text{rank}([\mathbf{x}_1, \ldots, \mathbf{x}_N]) \geq N$ then the training loss is 0!

Doesn't mean that the model will generalize to new samples!!!

What can we do?

Featurization

- $\mathbf{x}_n \leftarrow (1, x_n, x_n^2, x_n^3, \ldots, x_n^p) \in \mathbb{R}^{p+1}$
- Our $f(\mathbf{x}_n)$ is a polynomial of degree p
- We can perfectly fit p+1 points!



What can we do?

Deep Learnization

- Make f a nonlinear transformation of the input
- Wait, what?
- This time we don't prescribe the featurization process, it will be learned!

Doesn't mean that the model will generalize to new samples!!!

What can we do?

Featurization

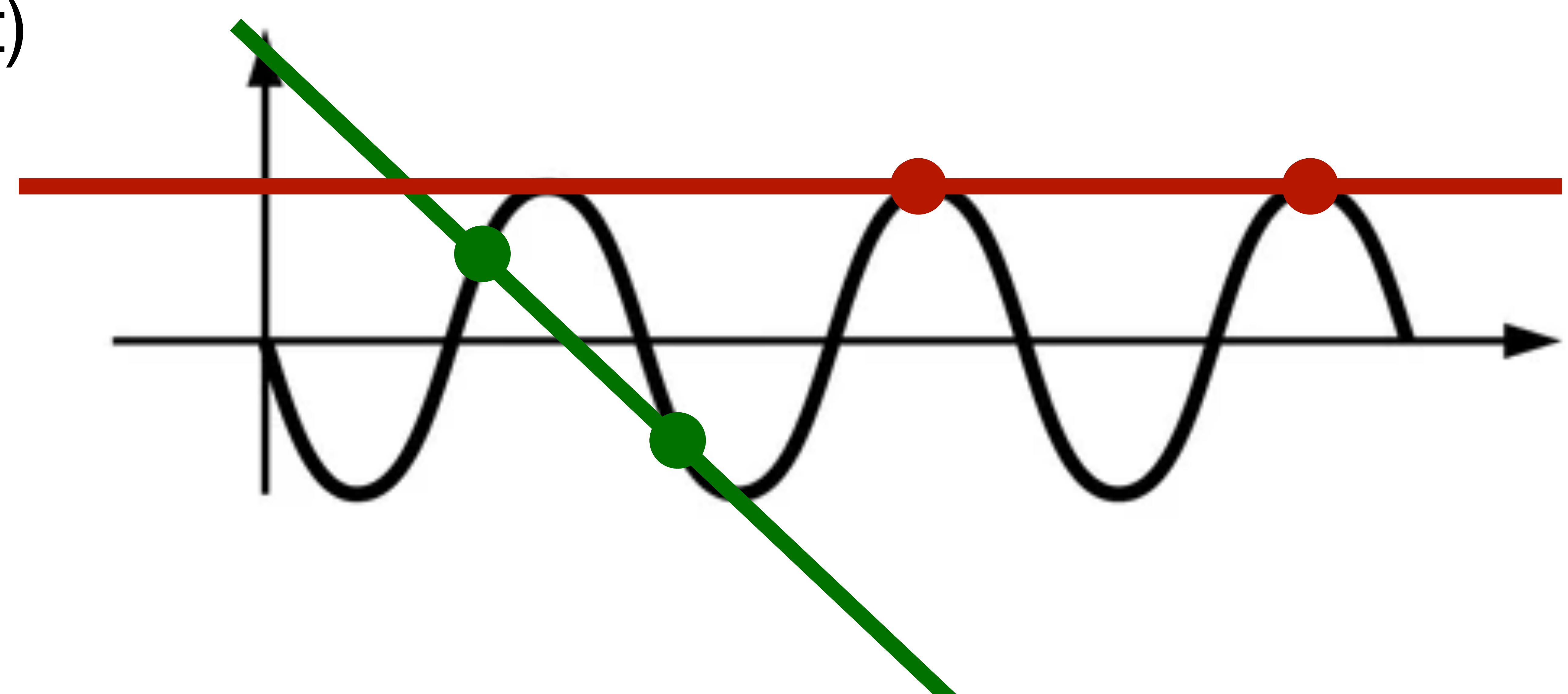
- Even a simple linear model….
- If you are in dimension $D = N$
- And points are not colinear
- Then training loss is 0 (perfect fit)

$$\sum_{n=1}^{N} (y_n - \langle \mathbf{w}, \mathbf{x}_n \rangle)^2 = \sum_{n=1}^{N} (y_n - \mathbf{x}_n^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y})^2 = \|\mathbf{y} - \mathbf{X}(\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}\|_F^2$$

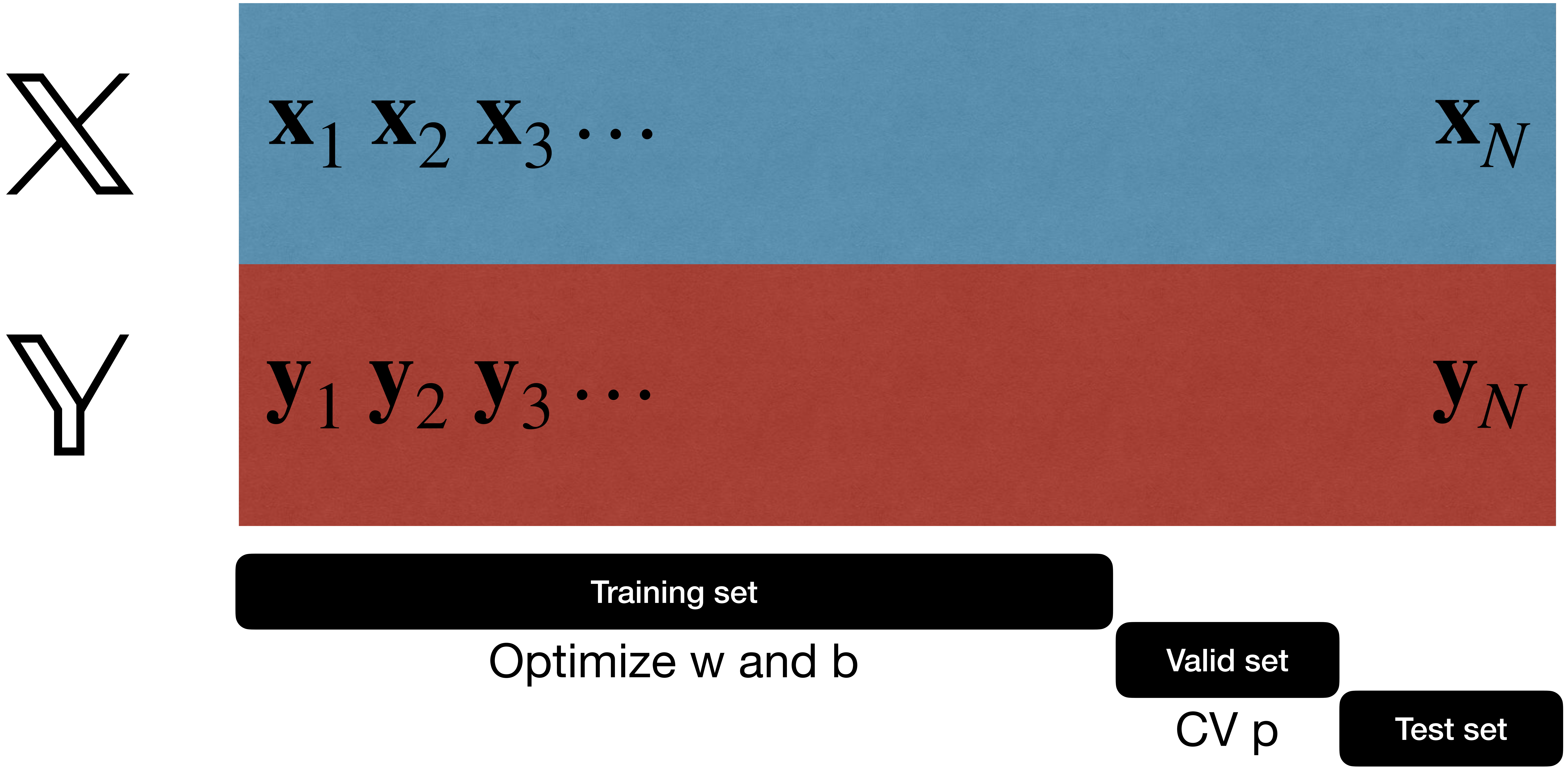
What can we do?

Featurization

- Even a simple linear model….
- If you are in dimension $D = N$ (or $D = N + 1$ with bias)
- And points are not colinear
- Then training loss is 0 (perfect fit)



How to evaluate f



How to evaluate f

How to evaluate f

- Typically one will use a 70/20/10 ratio

How to evaluate f

- Typically one will use a 70/20/10 ratio
- Can do many re-splits (K-fold cross-validation)

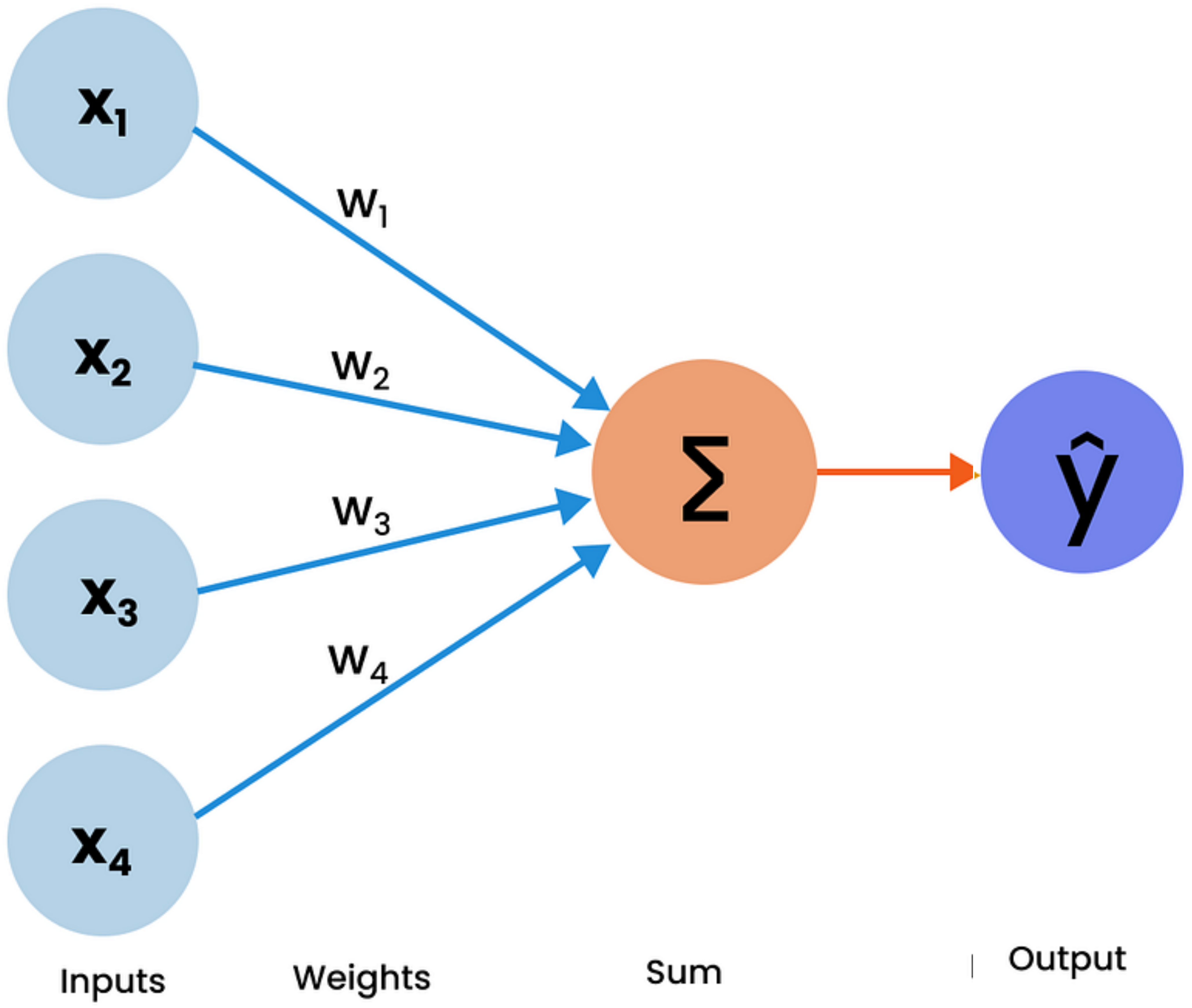
How to evaluate f

- Typically one will use a 70/20/10 ratio
- Can do many re-splits (K-fold cross-validation)
- The test set does inform about “in-distribution” generalization

How to evaluate f

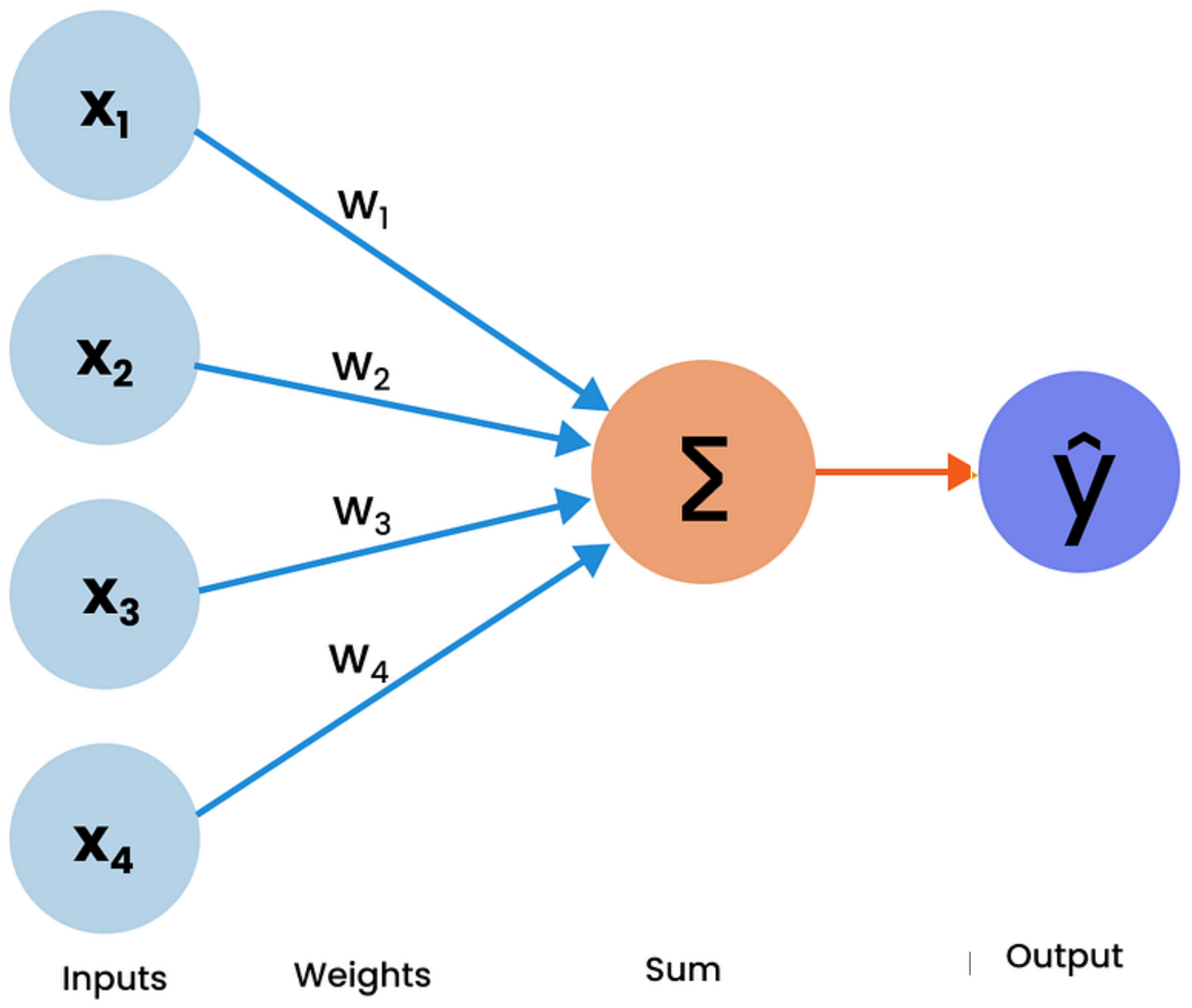
- Typically one will use a 70/20/10 ratio
- Can do many re-splits (K-fold cross-validation)
- The test set does inform about “in-distribution” generalization
- Deep Networks can have much more parameters than training samples without hurting generalization (main diff with other ML methods)

How to present f

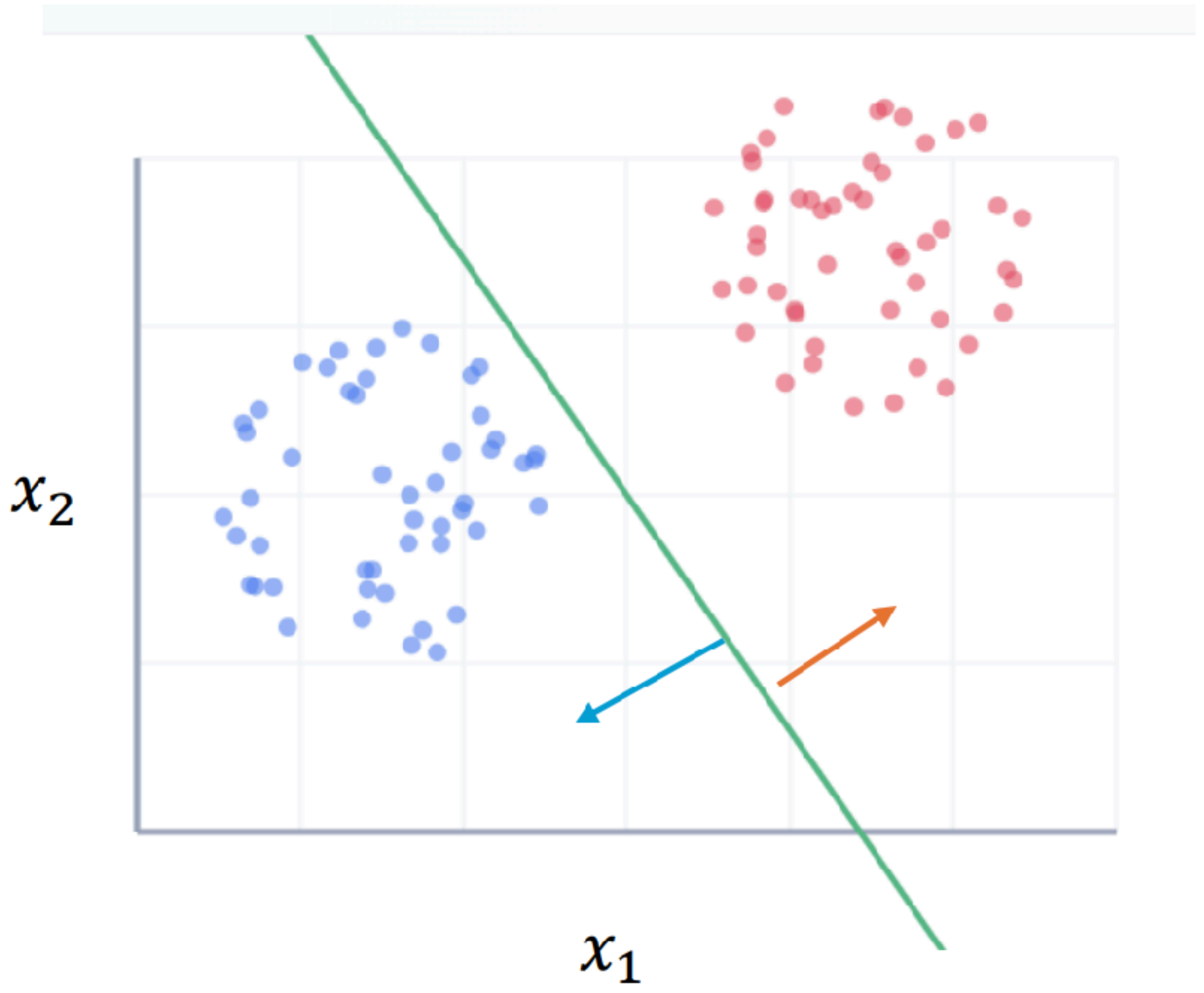


How to “interpret” f

- What if a weight is 0?
- What if a weight is much larger than others?



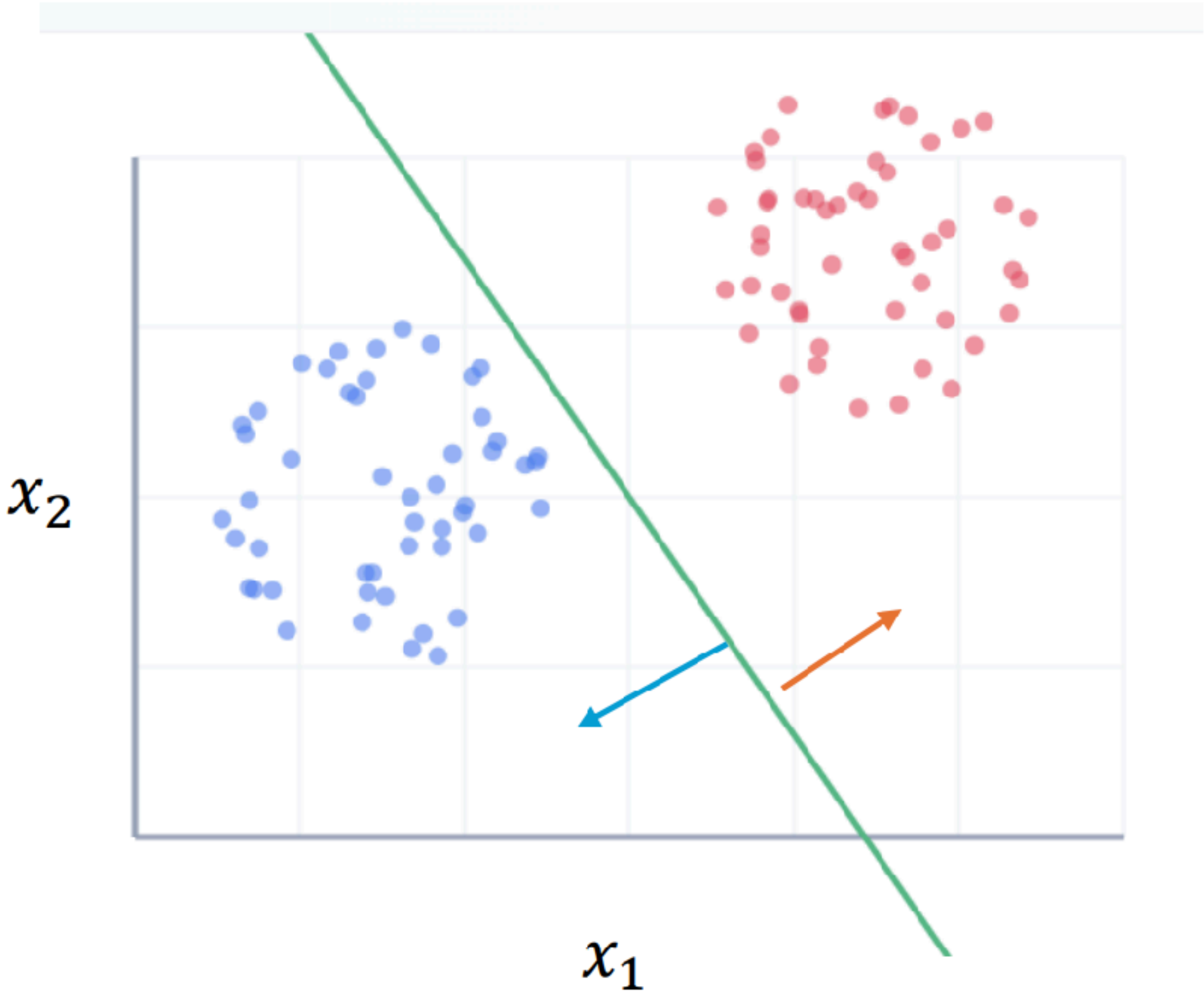
What about binary classification?

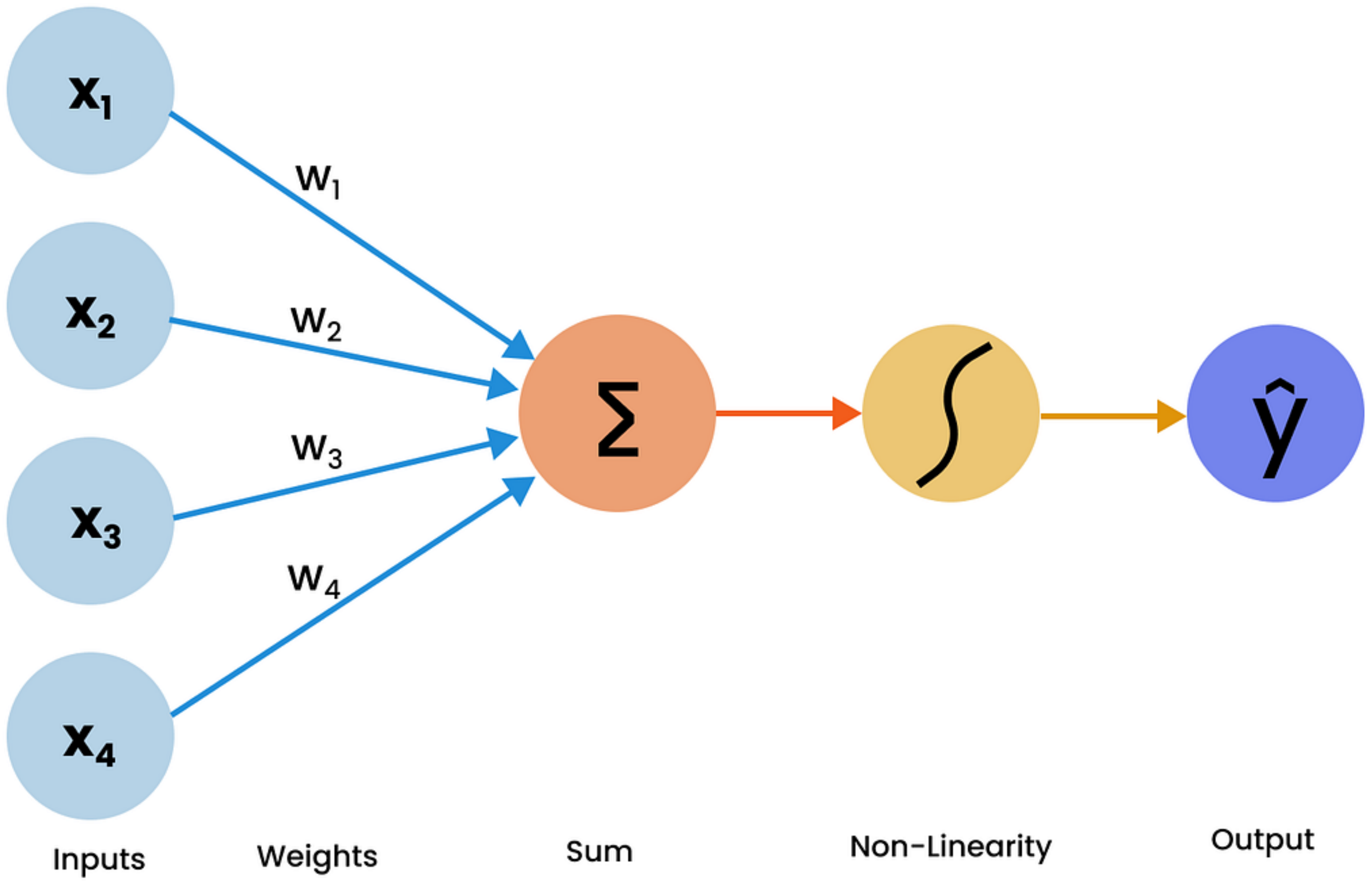


$$\hat{y} = w_1 \cdot x_1 + w_2 \cdot x_2 + b$$

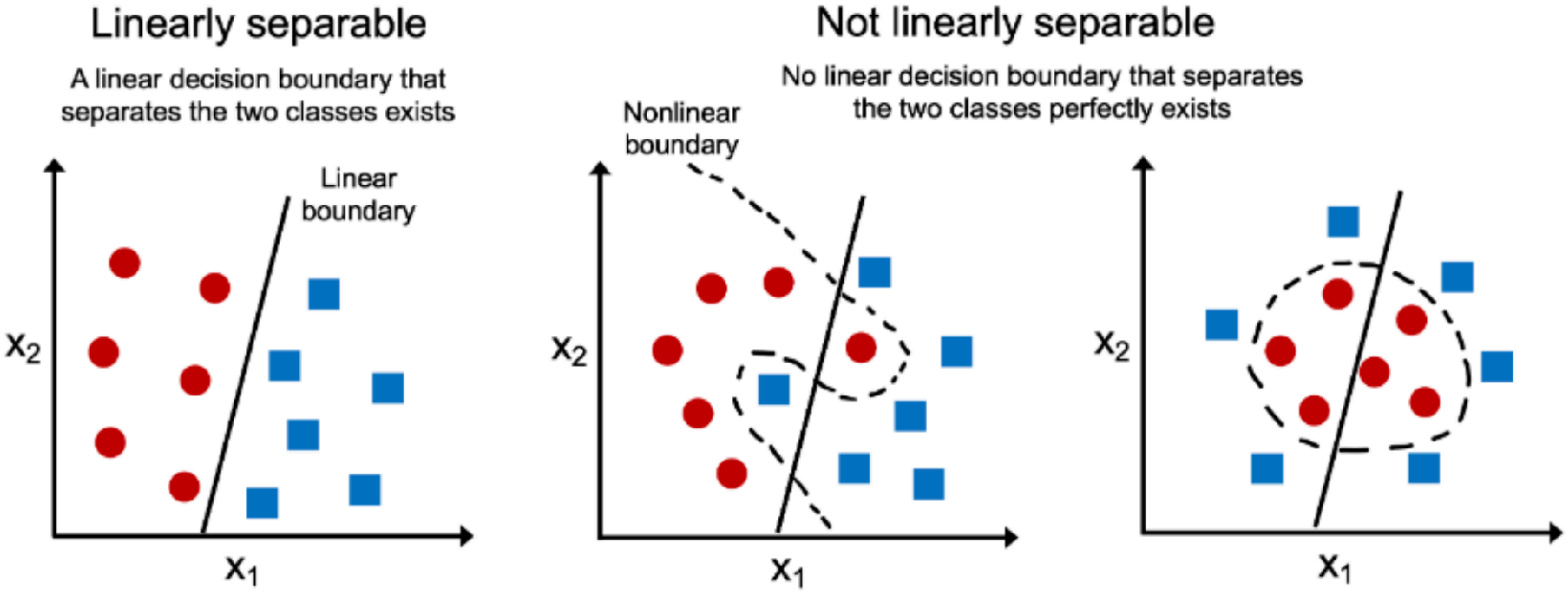
y is now interpreted as a probability

What about binary classification?





What about binary classification?



MNIST

The most famous dataset in Deep Learning

Modified **N**ational **I**nstitute of **S**tandards and **T**echnology database

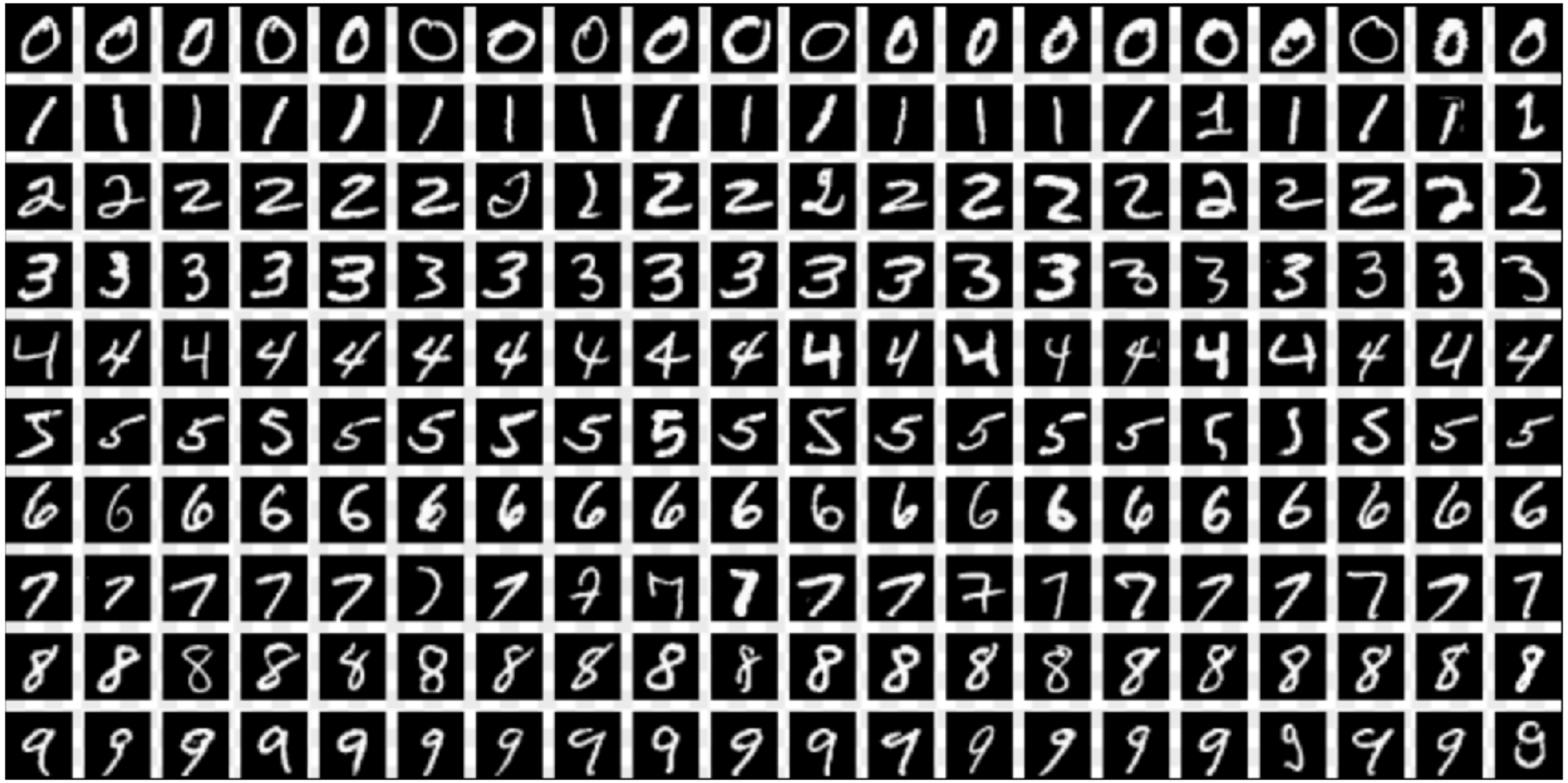
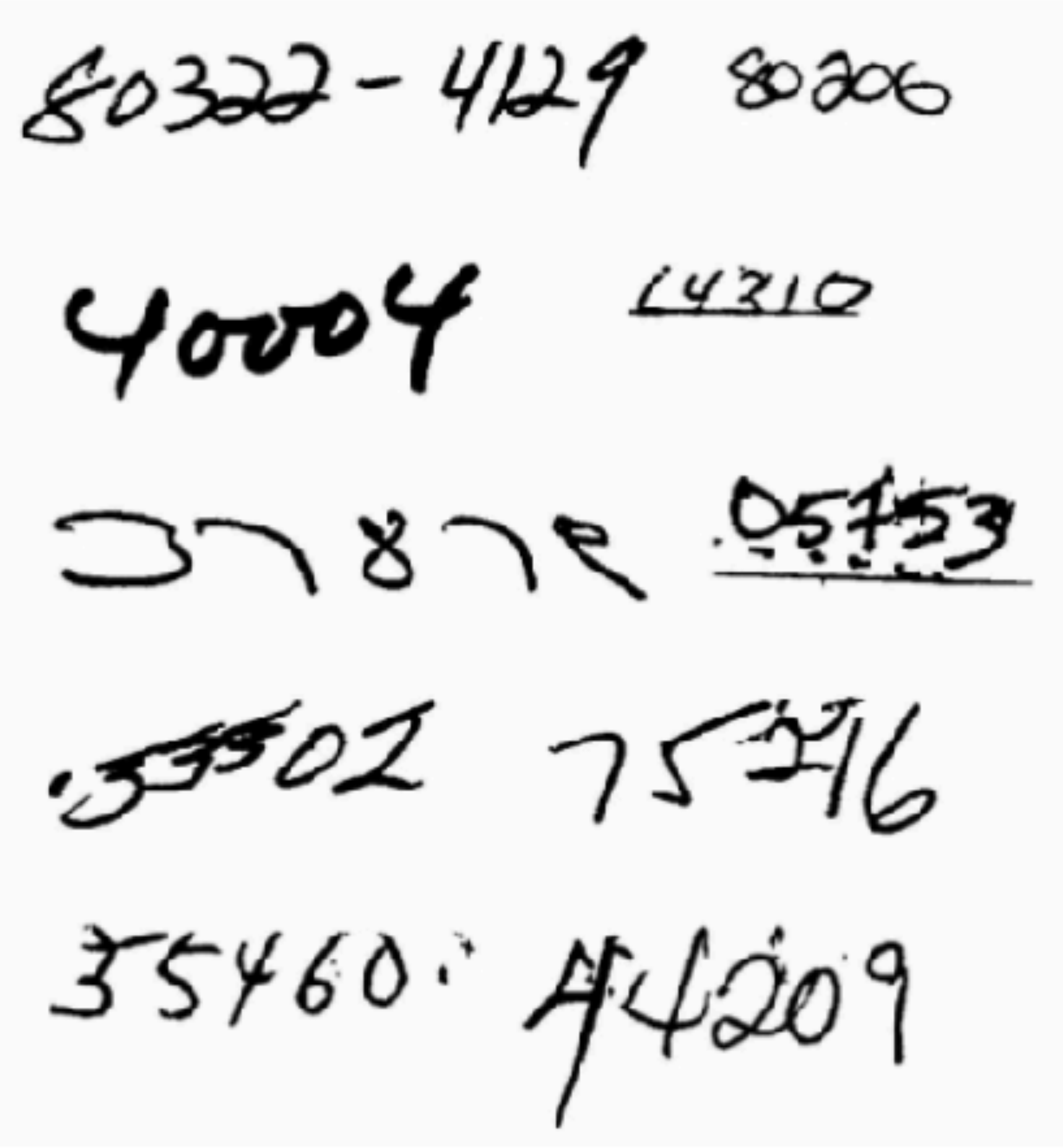


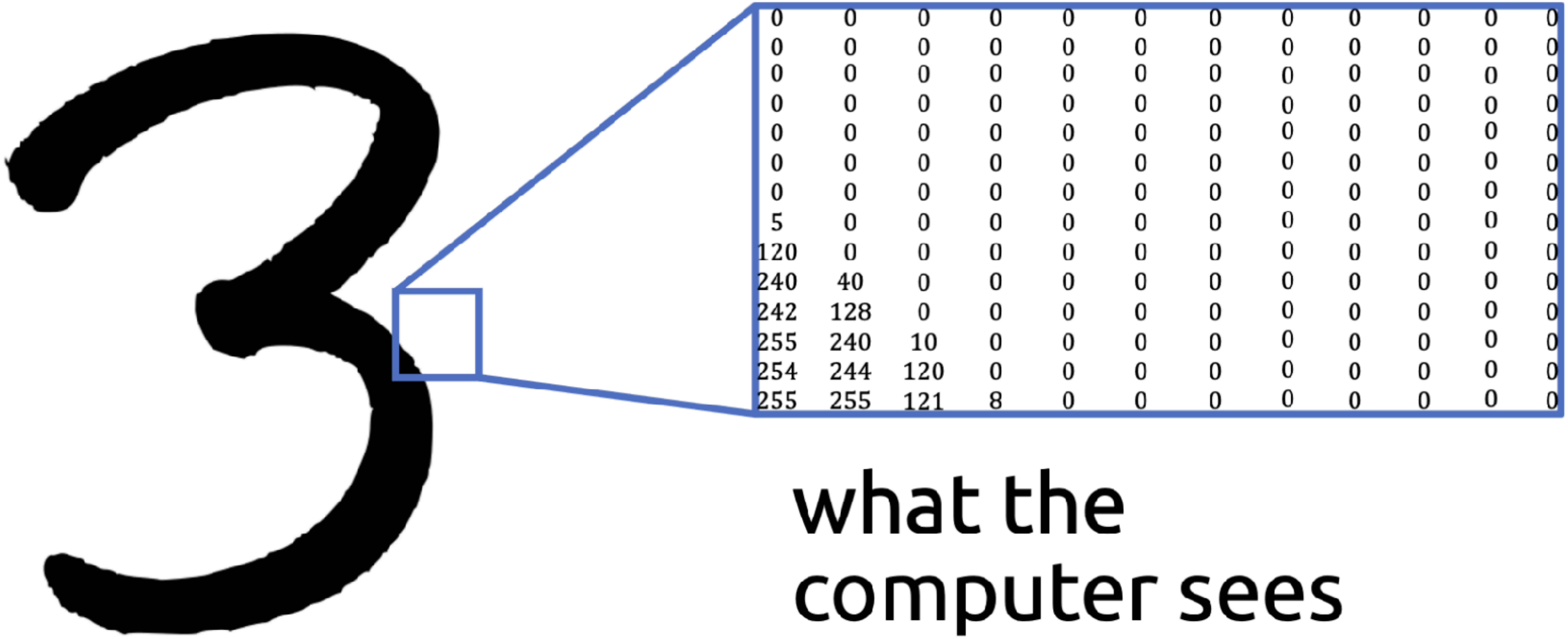
Image courtesy of Wikipedia

MNIST

- In 1990s, great increase in documents on paper (mail, checks, books, etc.)
- Motivation for a ZIP code recognizer on real U.S. mail for the postal service!



MNIST



what the computer sees

MNIST

Pixel Grid

$x^{(1)} =$ 

28x28 pixels

Function: f

$f(\mathbb{X}) \rightarrow \mathbb{Y}$

Which digit is it?

$y^{(1)} = \text{“2”}$

Questions?