Deep Learning (1470)

Randall Balestriero

Class 2

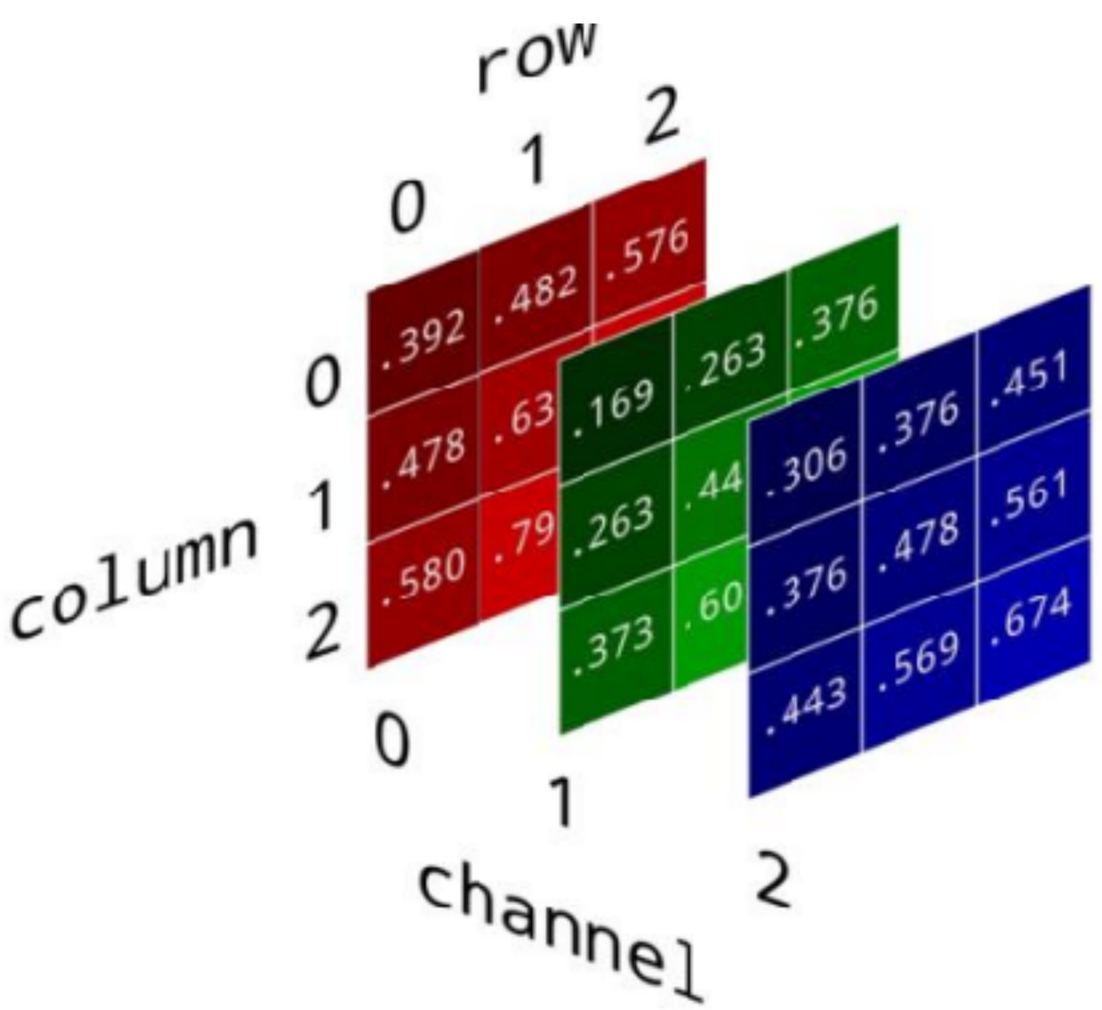
Rewind

Today's Roadmap

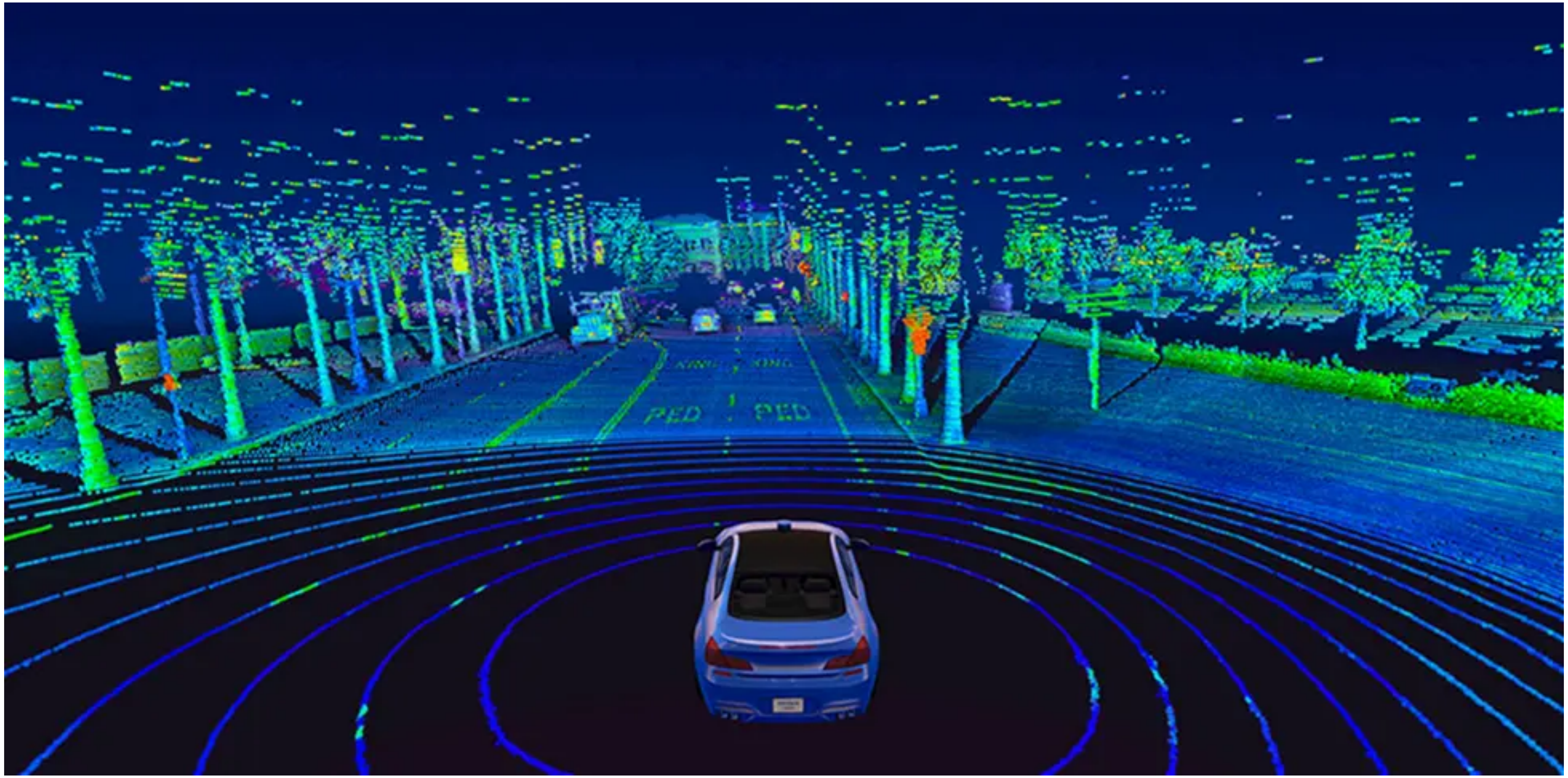
- How to represent inputs and outputs
- How to "train" f
- How to "evaluate" f

How to represent Inputs? (X)

How to represent Inputs? (X)



How to represent Inputs? (X)



(x, y, z, intensity) -> (x, y, z, D)

(VoxelNet)

How to represent Inputs? (X)

edges, vertices, features(v) (GNN)

How to represent Inputs? (X)

Nucleotides -> one-hot e.g. A -> (1, 0, 0, 0)

How to represent Outputs (y)

How to represent Outputs (y)

- Binary classification task?

How to represent Outputs (y)

- Binary classification task?
  - 0/1 or -1/1

How to represent Outputs (y)

- Binary classification task?
  - 0/1 or -1/1
- Multiclass classification task?

How to represent Outputs (y)

- Binary classification task?
  - 0/1 or -1/1
- Multiclass classification task?
  - 0/…/K-1

How to represent Outputs (y)

- Binary classification task?
  - 0/1 or -1/1
- Multiclass classification task?
  - 0/…/K-1
- Regression task?

How to represent Outputs (y)

- Binary classification task?
  - 0/1 or -1/1
- Multiclass classification task?
  - 0/…/K-1
- Regression task?
  - Actual value (standardized)

How to represent Outputs (y)

- Binary classification task?
  - 0/1 or -1/1
- Multiclass classification task?
  - 0/…/K-1
- Regression task?
  - Actual value (standardized)
  - -> bin and back to Multiclass classification task!

How to represent Outputs (y): Example

DeepMind

September 8, 2016 Research

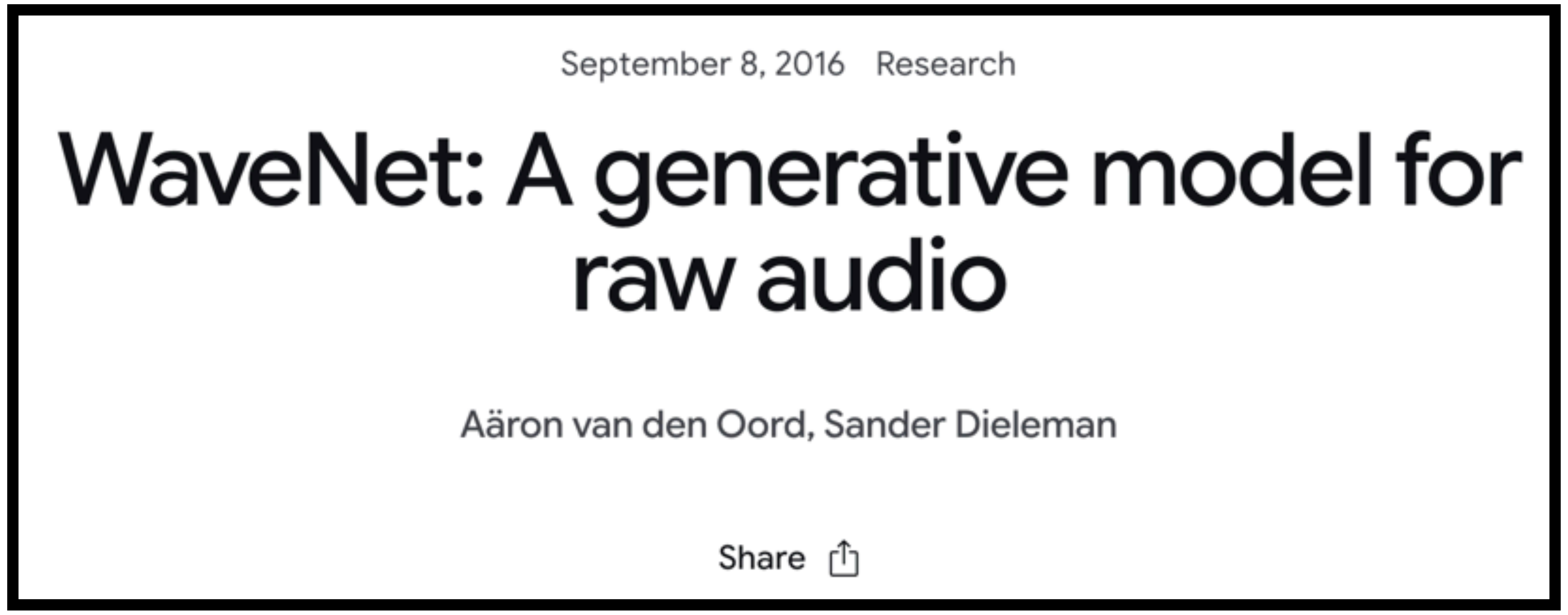
WaveNet: A generative model for raw audio

Aäron van den Oord, Sander Dieleman

Share

How to represent Outputs (y): Example

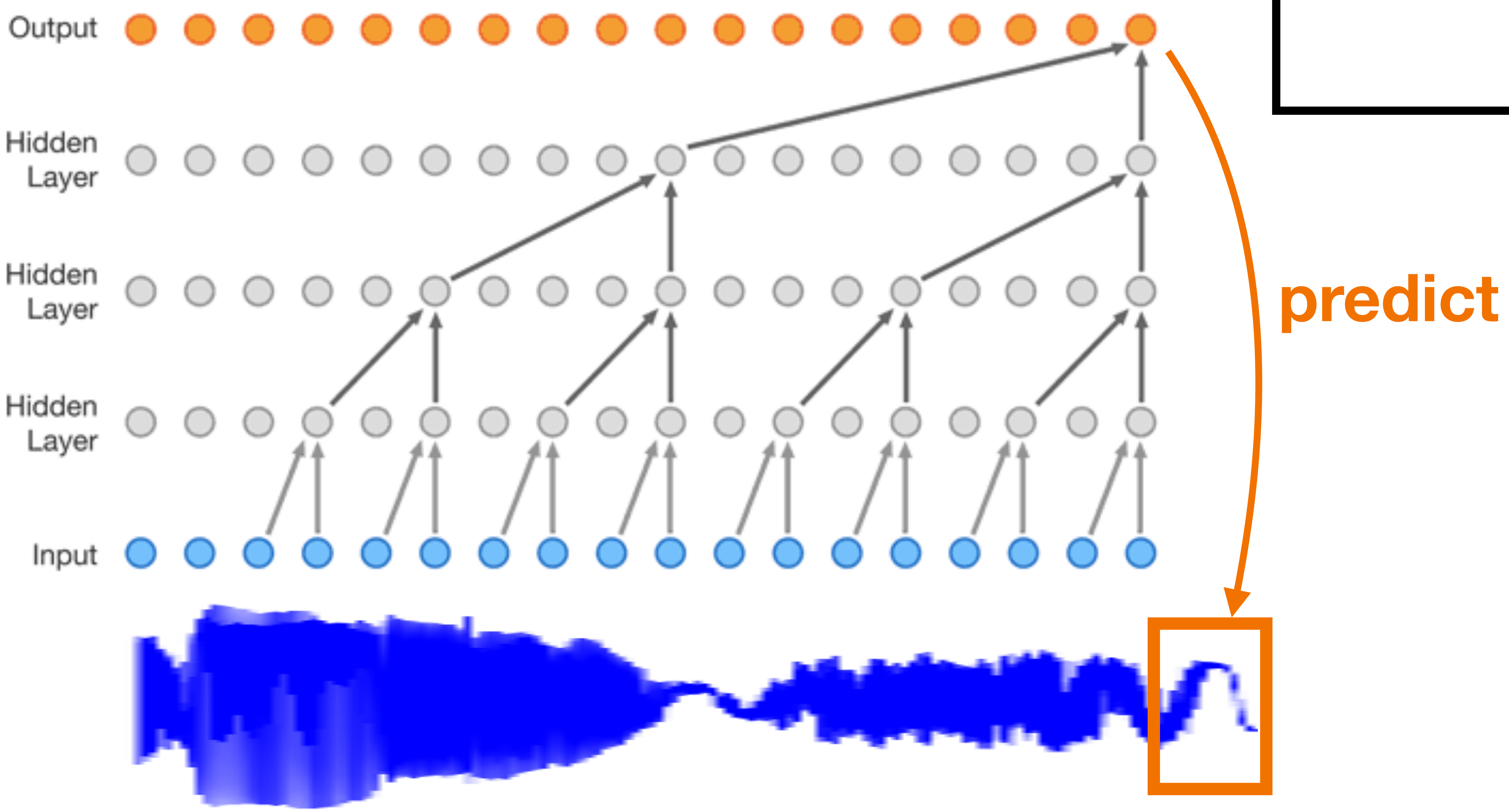
DeepMind


September 8, 2016 Research

WaveNet: A generative model for raw audio

Ääron van den Oord, Sander Dieleman

Share



How to represent Outputs (y): Example

DeepMind

September 8, 2016 Research

WaveNet: A generative model for raw audio

Aäron van den Oord, Sander Dieleman

Share

One approach to modeling the conditional distributions $p(x_t \mid x_1, \ldots, x_{t-1})$ over the individual audio samples would be to use a mixture model such as a mixture density network (Bishop, 1994) or mixture of conditional Gaussian scale mixtures (MCGSM) (Theis & Bethge, 2015). However, van den Oord et al. (2016a) showed that a softmax distribution tends to work better, even when the data is implicitly continuous (as is the case for image pixel intensities or audio sample values). One of the reasons is that a categorical distribution is more flexible and can more easily model arbitrary distributions because it makes no assumptions about their shape.

Because raw audio is typically stored as a sequence of 16-bit integer values (one per timestep), a softmax layer would need to output 65,536 probabilities per timestep to model all possible values. To make this more tractable, we first apply a μ-law companding transformation (ITU-T, 1988) to the data, and then quantize it to 256 possible values:

Notations

Notations

- $\mathbb{R}$: the set of real numbers

Notations

- $\mathbb{R}$: the set of real numbers
- $\mathbf{v} \in \mathbb{R}^D$: a D-dimensional **vector**

Notations

- $\mathbb{R}$: the set of real numbers
- $\mathbf{v} \in \mathbb{R}^{D}$: a D-dimensional **vector**
- $\mathbf{V} \in \mathbb{R}^{K \times D}$: a K-by-D **matrix**

Notations

- $\mathbb{R}$: the set of real numbers
- $\mathbf{v} \in \mathbb{R}^D$: a D-dimensional **vector**
- $\mathbf{V} \in \mathbb{R}^{K \times D}$: a K-by-D **matrix**
- $\mathbf{V} \in \mathbb{R}^{C \times K \times D}$: a C-by-K-by-D **tensor**

Notations

Do you have an example (from previous slides)?

- $\mathbb{R}$: the set of real numbers
- $\mathbf{v} \in \mathbb{R}^{D}$: a D-dimensional **vector**
- $\mathbf{V} \in \mathbb{R}^{K \times D}$: a K-by-D **matrix**
- $\mathbf{V} \in \mathbb{R}^{C \times K \times D}$: a C-by-K-by-D **tensor**

Notations

Do you have an example (from previous slides)?

- $\mathbb{R}$: the set of real numbers
- $\mathbf{v} \in \mathbb{R}^{D}$: a D-dimensional **vector**
- $\mathbf{V} \in \mathbb{R}^{K \times D}$: a K-by-D **matrix**
- $\mathbf{V} \in \mathbb{R}^{C \times K \times D}$: a C-by-K-by-D **tensor**
- $\mathbb{X} \triangleq \left(\mathbf{x}_1, \ldots, \mathbf{x}_N\right)$: the input samples

Notations

Do you have an example (from previous slides)?

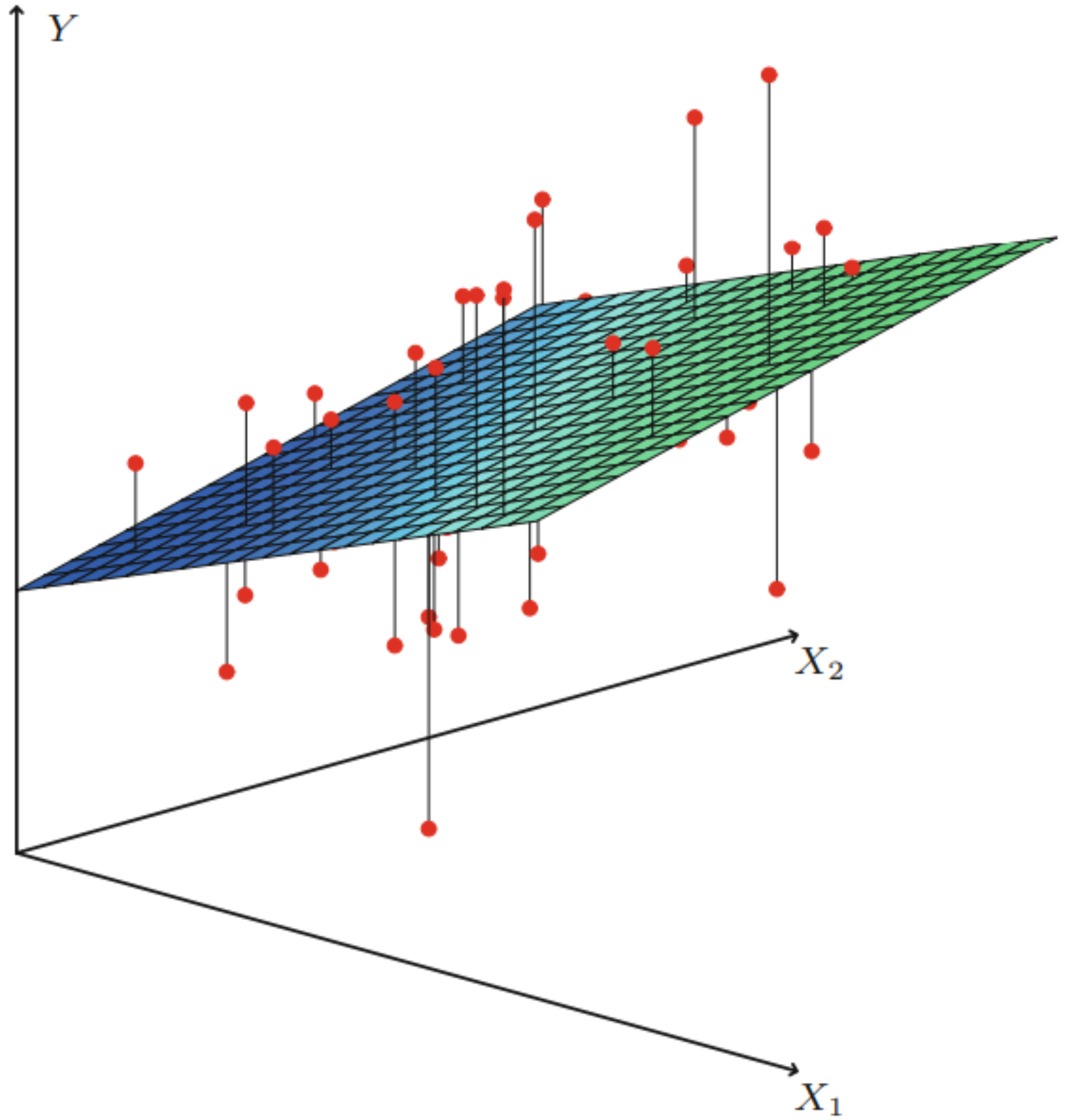
- $\mathbb{R}$: the set of real numbers
- $\mathbf{v} \in \mathbb{R}^{D}$: a D-dimensional **vector**
- $\mathbf{V} \in \mathbb{R}^{K \times D}$: a K-by-D **matrix**
- $\mathbf{V} \in \mathbb{R}^{C \times K \times D}$: a C-by-K-by-D **tensor**
- $\mathbb{X} \triangleq \left(\mathbf{x}_1, \ldots, \mathbf{x}_N\right)$: the input samples
- $\mathbb{Y} \triangleq \left(\mathbf{y}_1, \ldots, \mathbf{y}_N\right)$: the output samples

How to train f

- Let's start with a simple model: $f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$

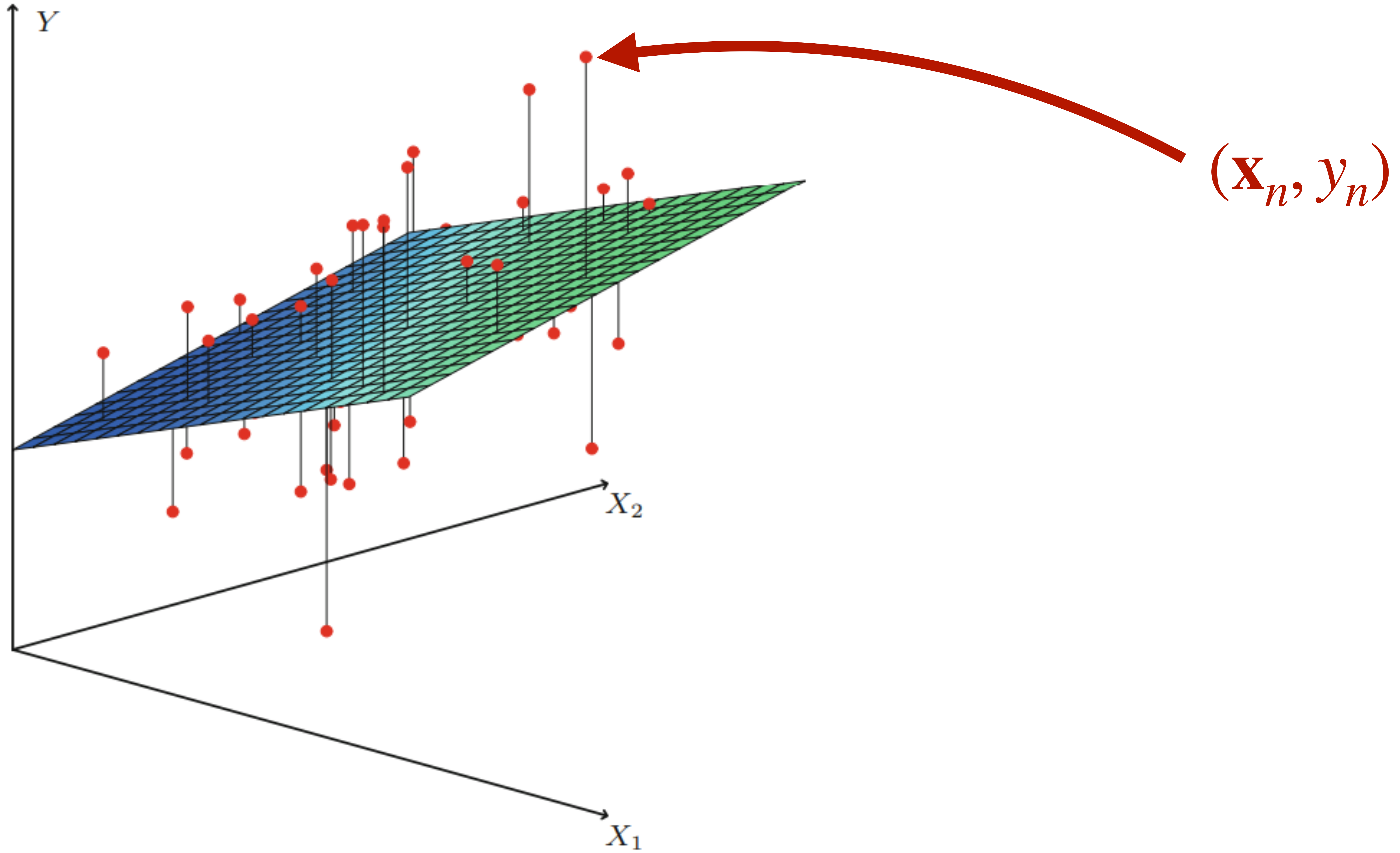
How to train f

- Let's start with a simple model: $f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$



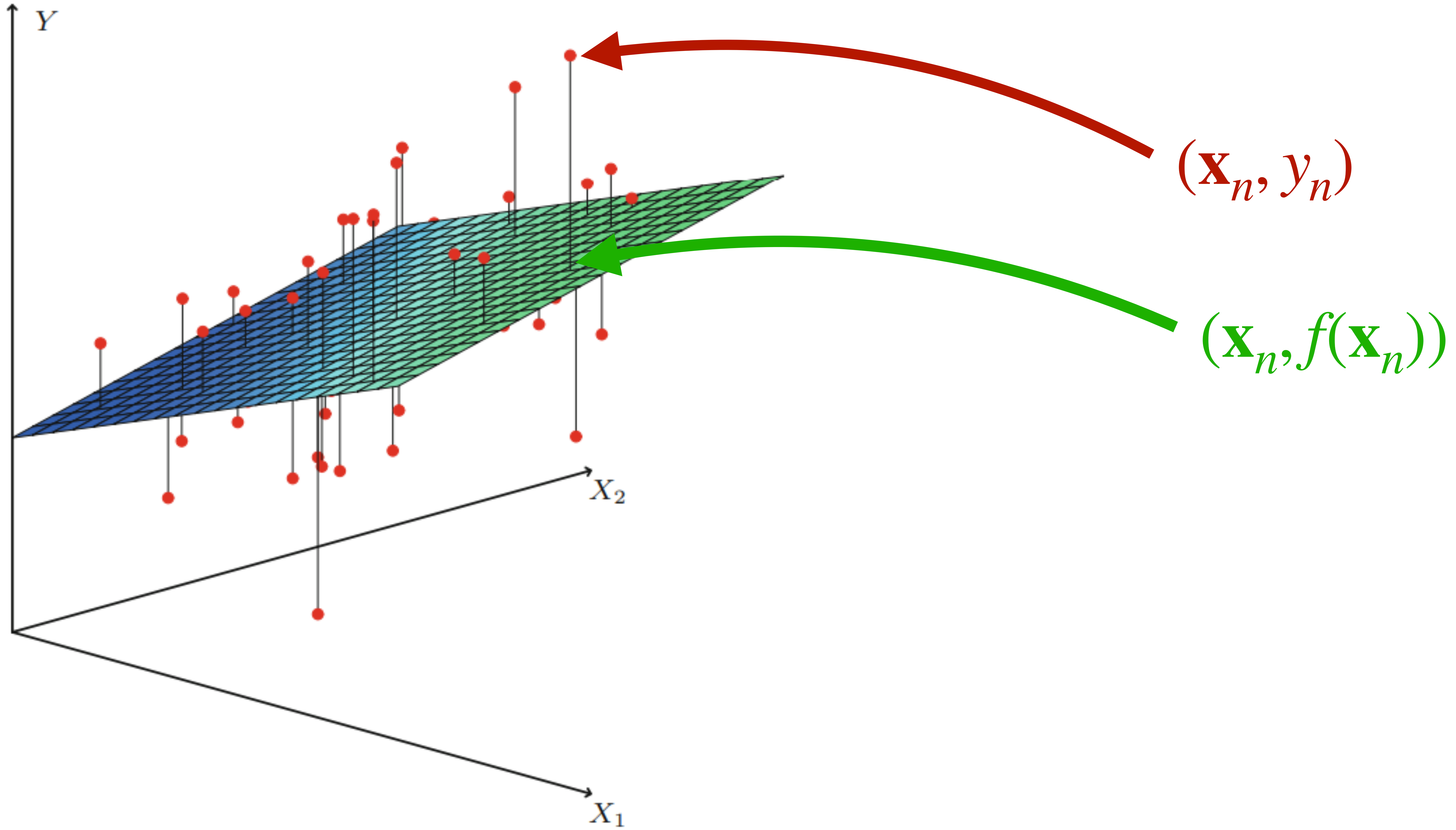
How to train f

- Let's start with a simple model: $f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$



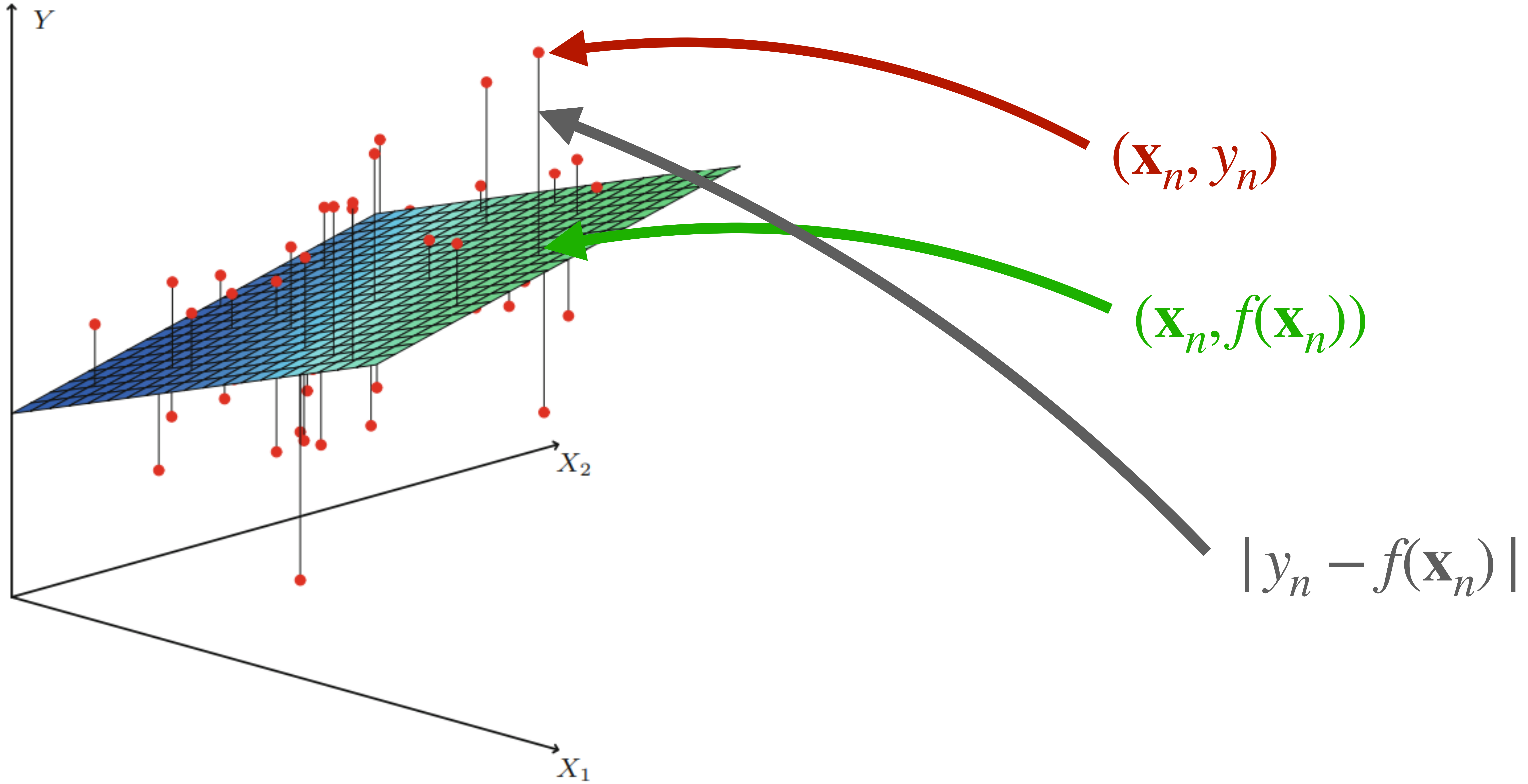
How to train f

- Let's start with a simple model: $f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$



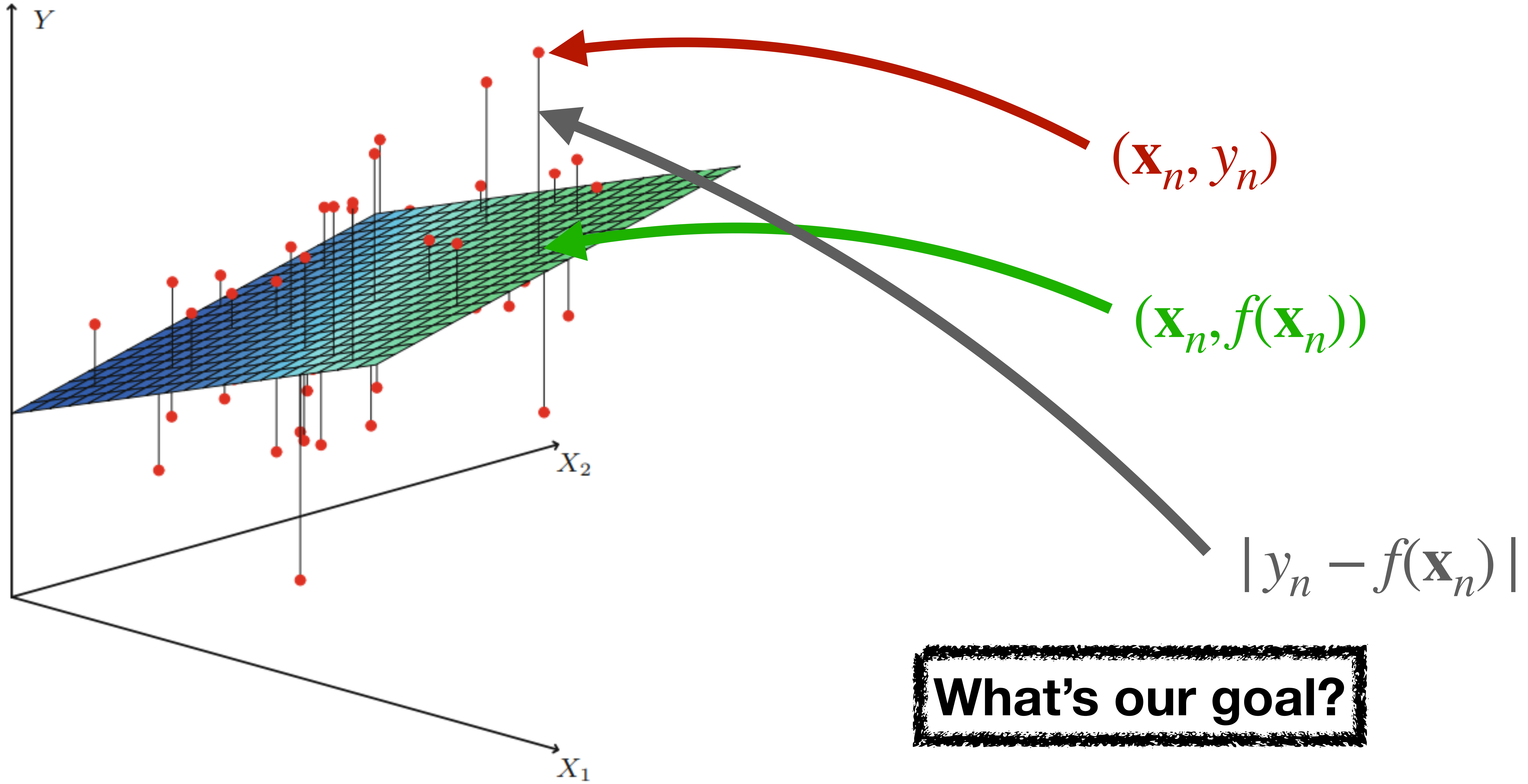
How to train f

- Let's start with a simple model: $f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$



How to train f

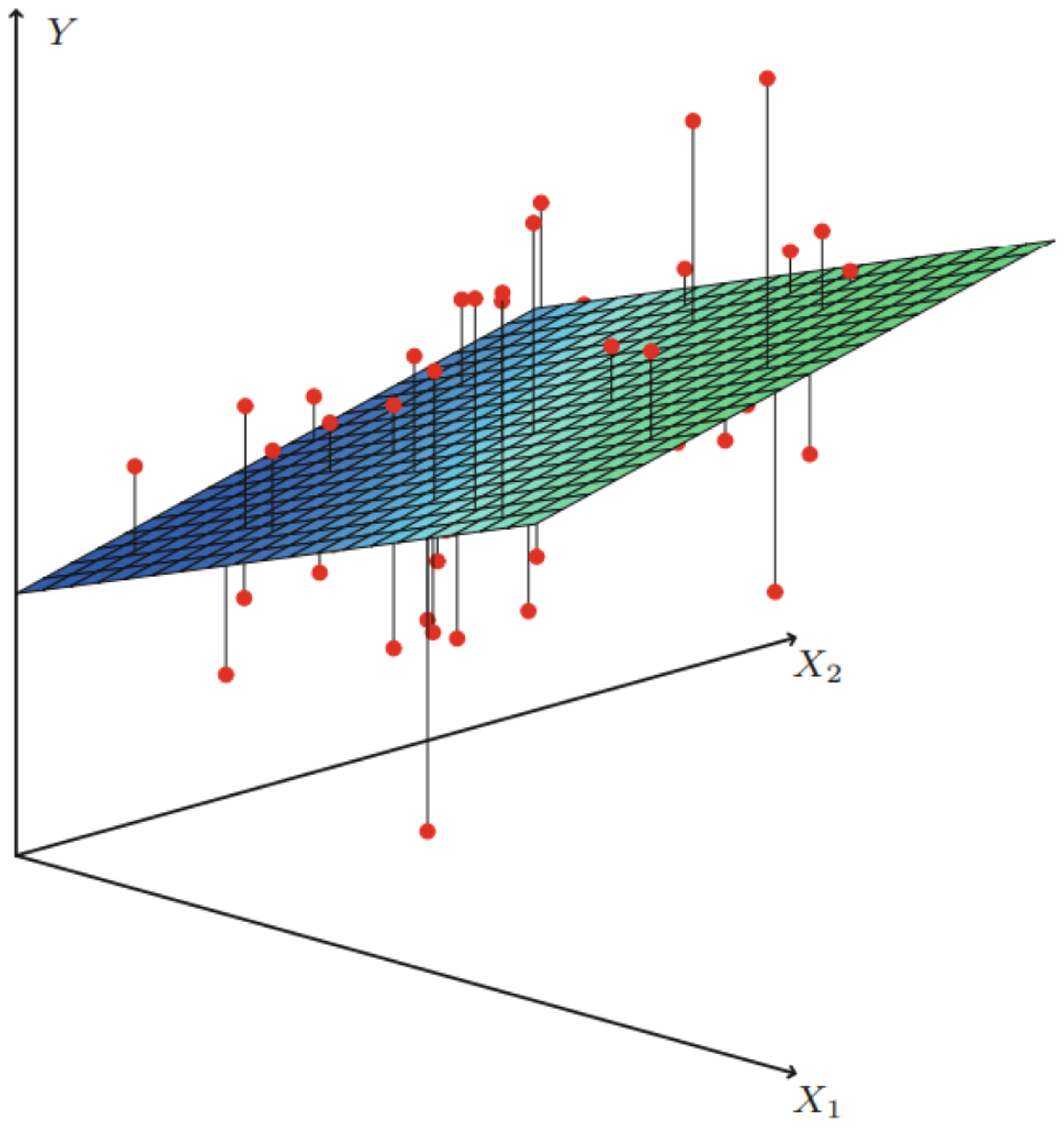
- Let's start with a simple model: $f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$



What's our goal?

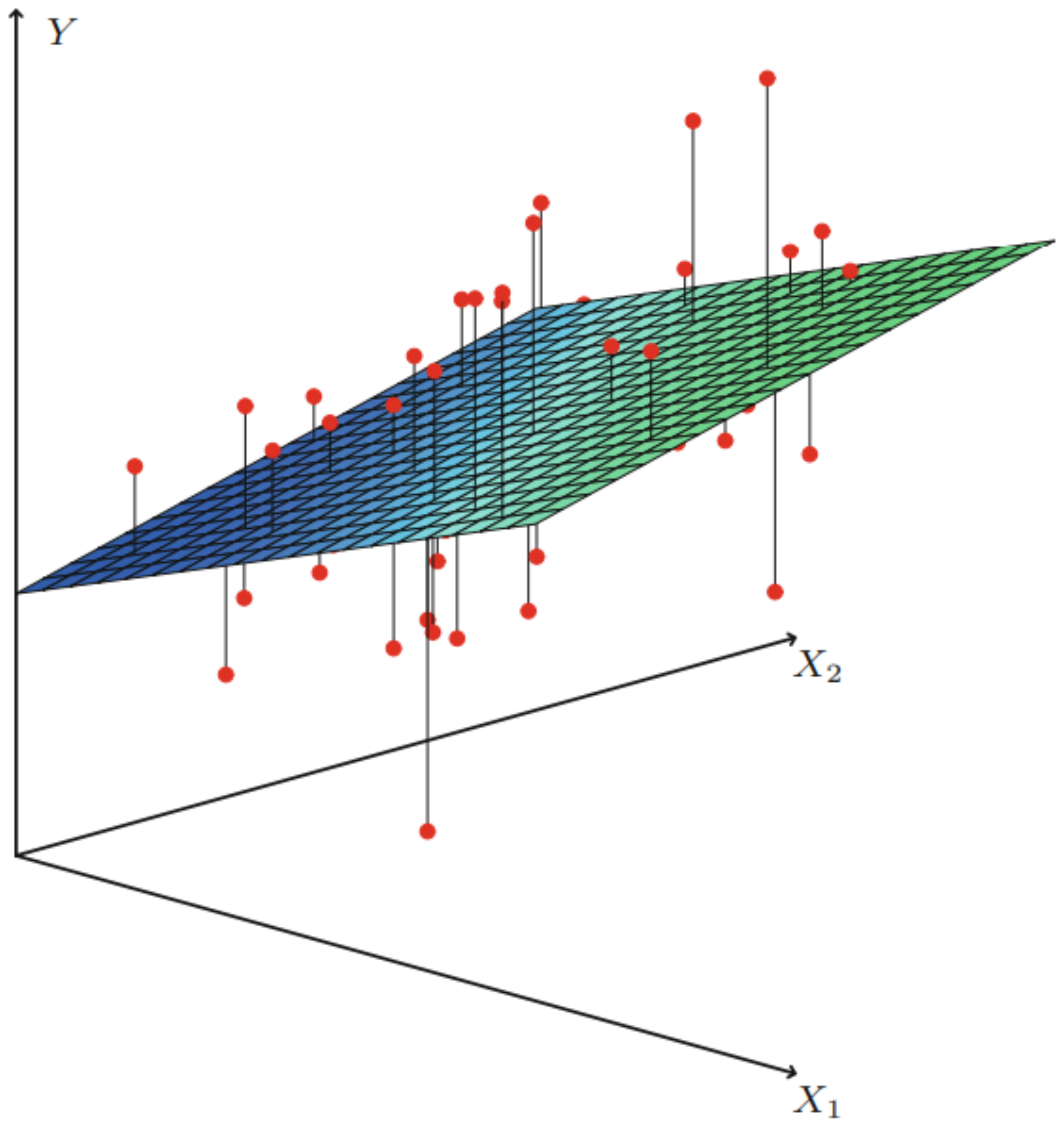
How to train f

- Let's start with a simple model: $f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$
- $\min_{\mathbf{w},b} \sum_{n=1}^{N} |y_n - f(\mathbf{x}_n)| = \min_{\mathbf{w},b} \sum_{n=1}^{N} |y_n - \langle \mathbf{w}, \mathbf{x}_n \rangle - b|$



How to train f

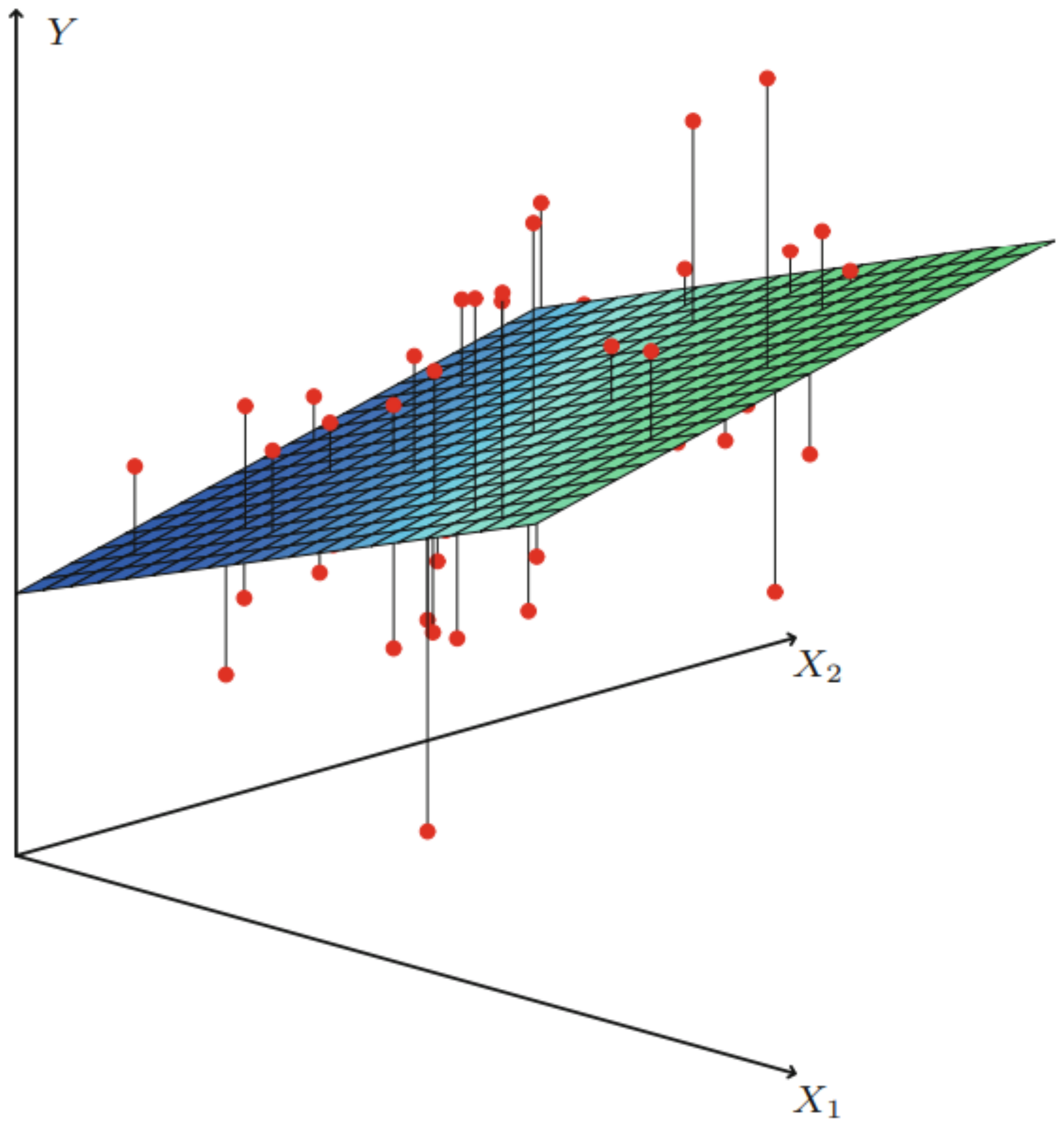
- Let's start with a simple model: $f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$
- $\min_{\mathbf{w},b} \sum_{n=1}^{N} |y_n - f(\mathbf{x}_n)| = \min_{\mathbf{w},b} \sum_{n=1}^{N} |y_n - \langle \mathbf{w}, \mathbf{x}_n \rangle - b|$



How to "optimize" the parameters?

How to train f

- Let's start with a simple model: $f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$
- $\min_{\mathbf{w},b} \sum_{n=1}^{N} | y_n - f(\mathbf{x}_n) | = \min_{\mathbf{w},b} \sum_{n=1}^{N} | y_n - \langle \mathbf{w}, \mathbf{x}_n \rangle - b |$

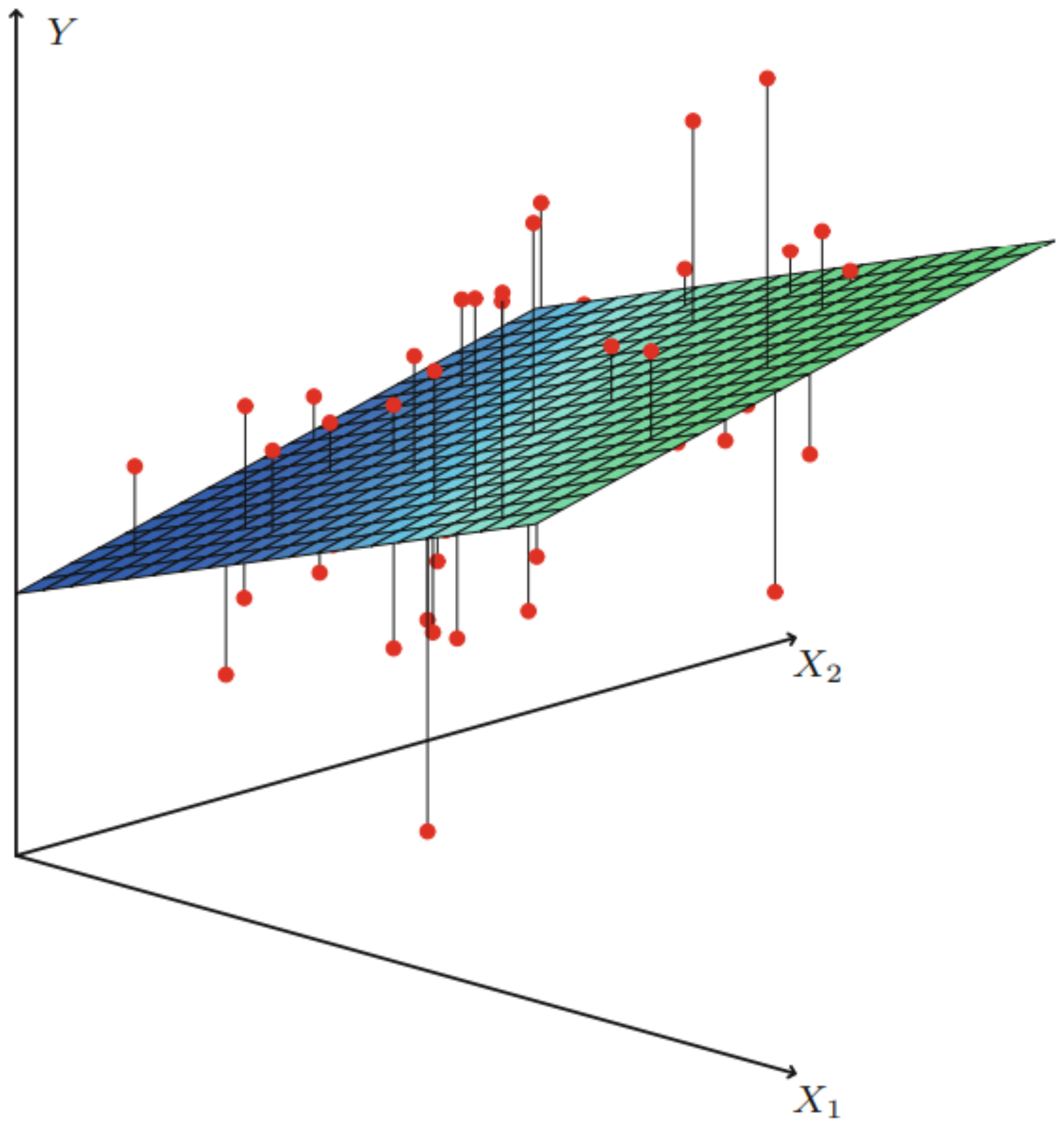


How to "optimize" the parameters?

- Random search (curse of dimensionality)

How to train f

- Let's start with a simple model: $f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$
- $\min_{\mathbf{w},b} \sum_{n=1}^{N} |y_n - f(\mathbf{x}_n)| = \min_{\mathbf{w},b} \sum_{n=1}^{N} |y_n - \langle \mathbf{w}, \mathbf{x}_n \rangle - b|$

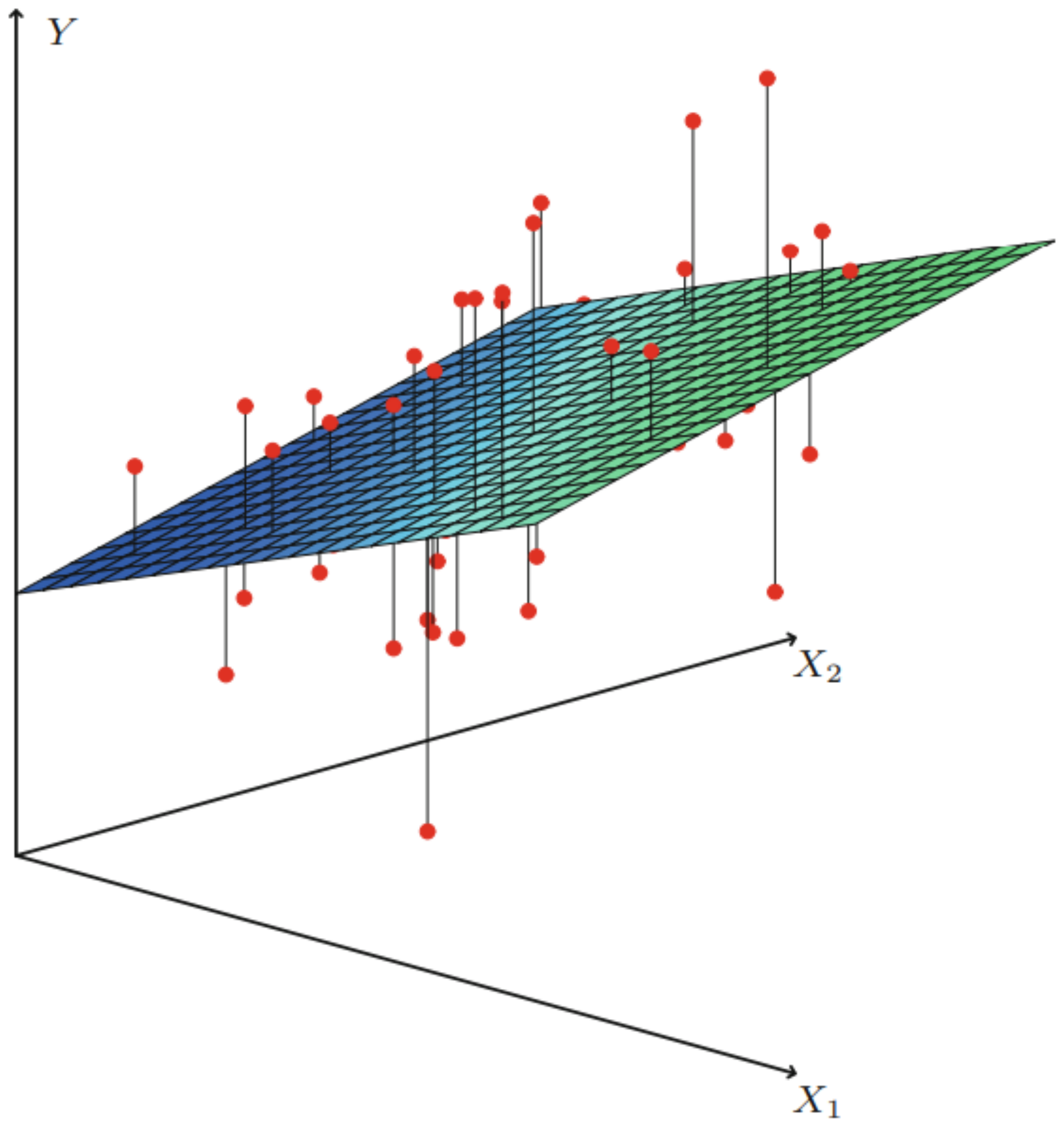


$\boxed{\textbf{How to “optimize” the parameters?}}$

- Random search (curse of dimensionality)
- Closed-form (only in a few cases)

How to train f

- Let's start with a simple model: $f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$
- $\min_{\mathbf{w},b} \sum_{n=1}^{N} |y_n - f(\mathbf{x}_n)| = \min_{\mathbf{w},b} \sum_{n=1}^{N} |y_n - \langle \mathbf{w}, \mathbf{x}_n \rangle - b|$

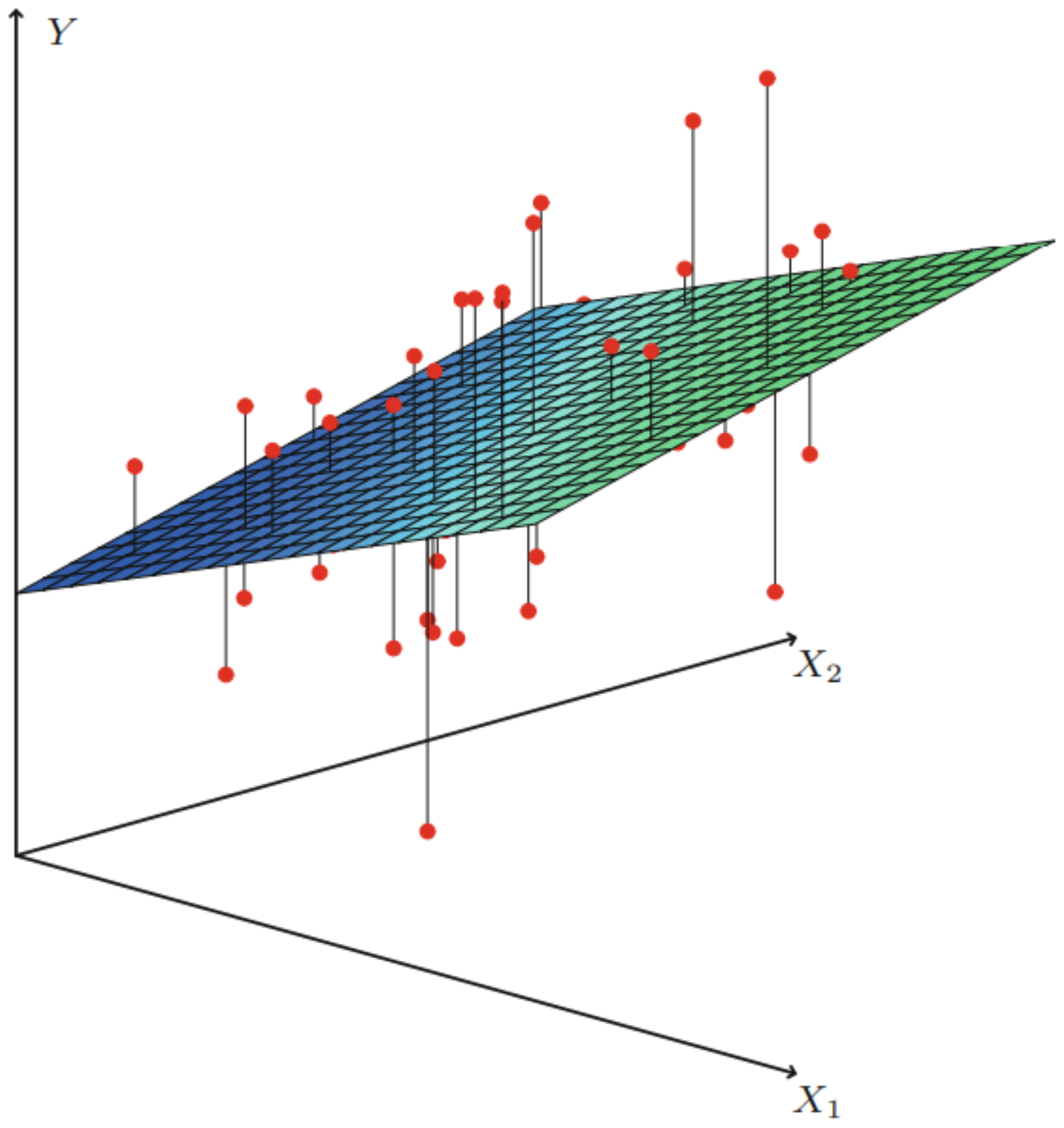


How to "optimize" the parameters?

- Random search (curse of dimensionality)
- Closed-form (only in a few cases)
- Gradient-based optimization

How to train f

- Let's start with a simple model: $f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$
- $\min_{\mathbf{w},b} \sum_{n=1}^{N} | y_n - f(\mathbf{x}_n) | = \min_{\mathbf{w},b} \sum_{n=1}^{N} | y_n - \langle \mathbf{w}, \mathbf{x}_n \rangle - b |$

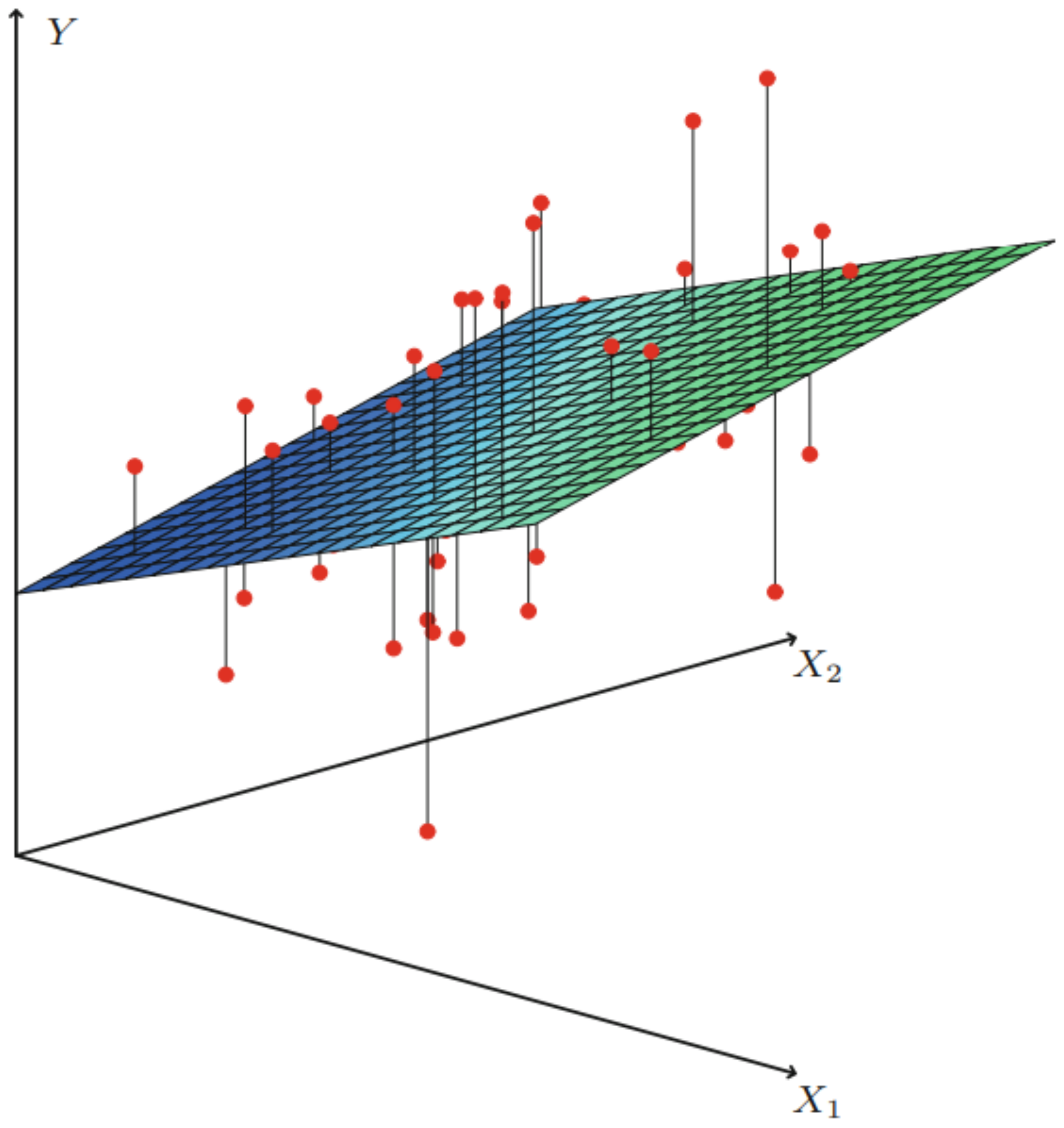


How to train f

- Let's start with a simple model: $f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$
- $\min_{\mathbf{w},b} \sum_{n=1}^{N} |y_n - f(\mathbf{x}_n)| = \min_{\mathbf{w},b} \sum_{n=1}^{N} |y_n - \langle \mathbf{w}, \mathbf{x}_n \rangle - b|$

or

$$|y_n - \langle \mathbf{w}, \mathbf{x}_n \rangle - b|^2$$



How to train f

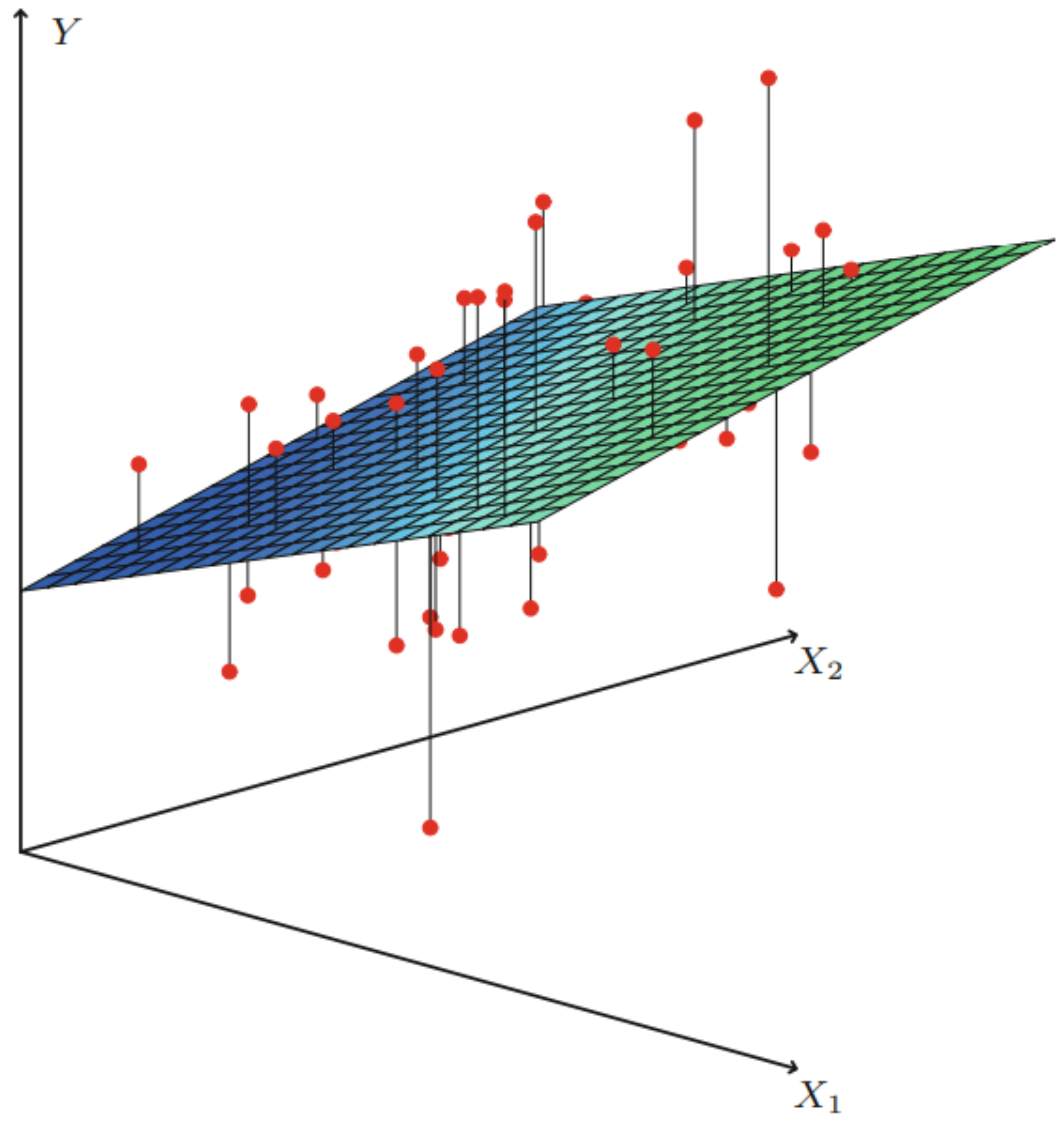
- Let's start with a simple model: $f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$
- $\min_{\mathbf{w},b} \sum_{n=1}^{N} |y_n - f(\mathbf{x}_n)| = \min_{\mathbf{w},b} \sum_{n=1}^{N} |y_n - \langle \mathbf{w}, \mathbf{x}_n \rangle - b|$

or

$$|y_n - \langle \mathbf{w}, \mathbf{x}_n \rangle - b|^2$$

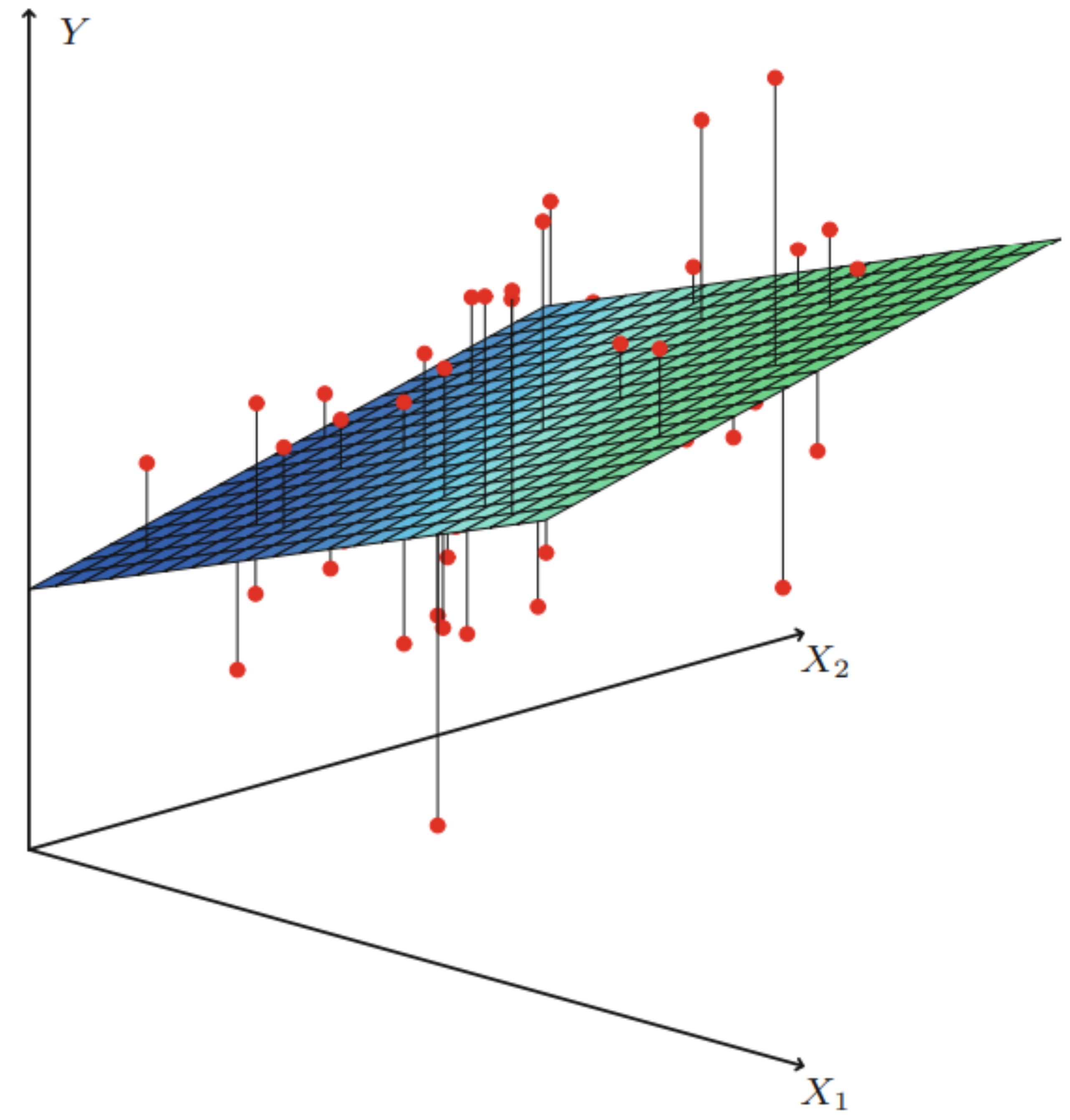
or

$$|y_n - \langle \mathbf{w}, \mathbf{x}_n \rangle - b|^{0.5}$$



How to train f

- Let's start with a simple model: $f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$
- $\min_{\mathbf{w},b} \sum_{n=1}^{N} |y_n - f(\mathbf{x}_n)| = \min_{\mathbf{w},b} \sum_{n=1}^{N} |y_n - \langle \mathbf{w}, \mathbf{x}_n \rangle - b|$



or

$|y_n - \langle \mathbf{w}, \mathbf{x}_n \rangle - b|^2$

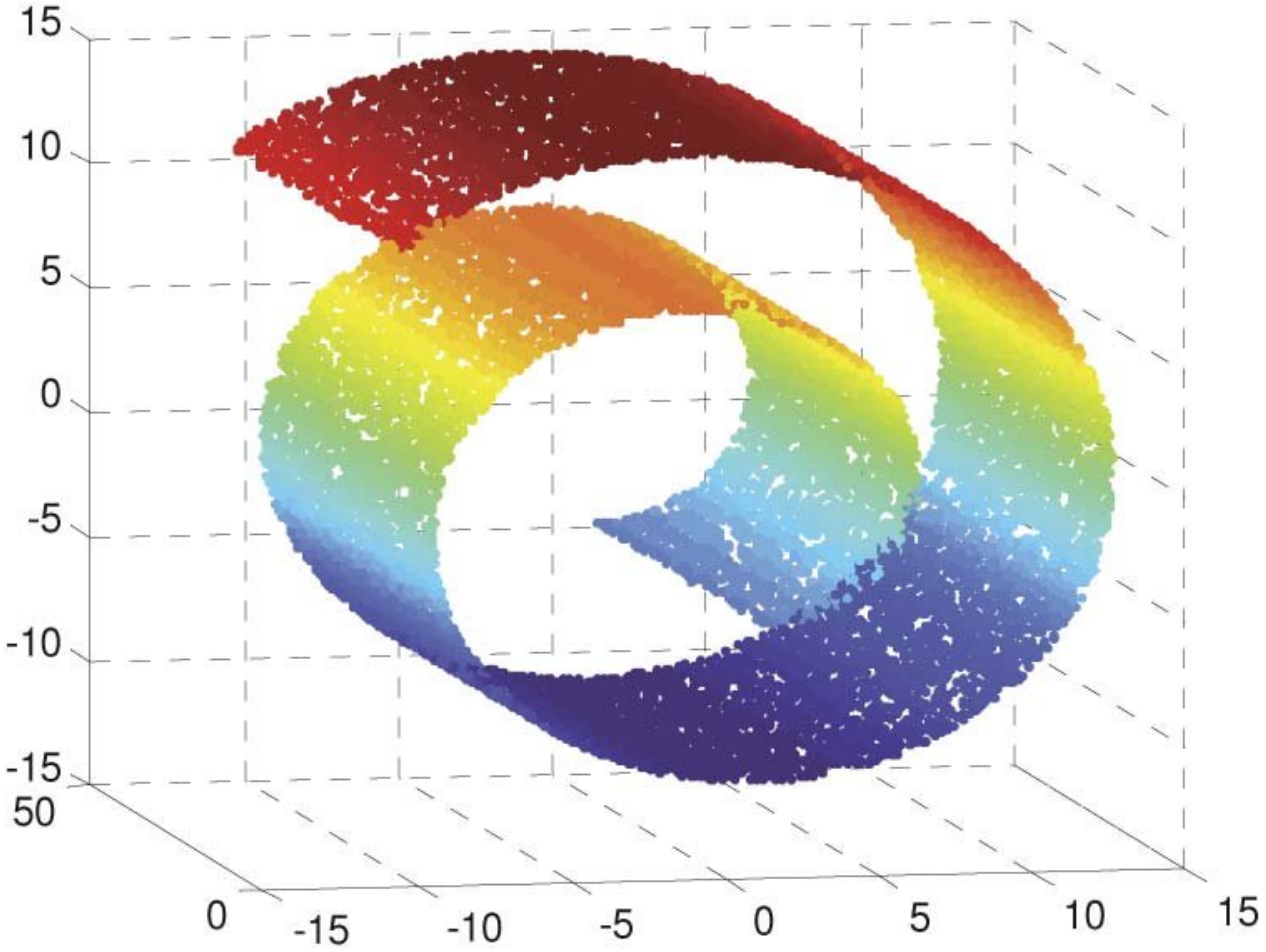
MSE

or

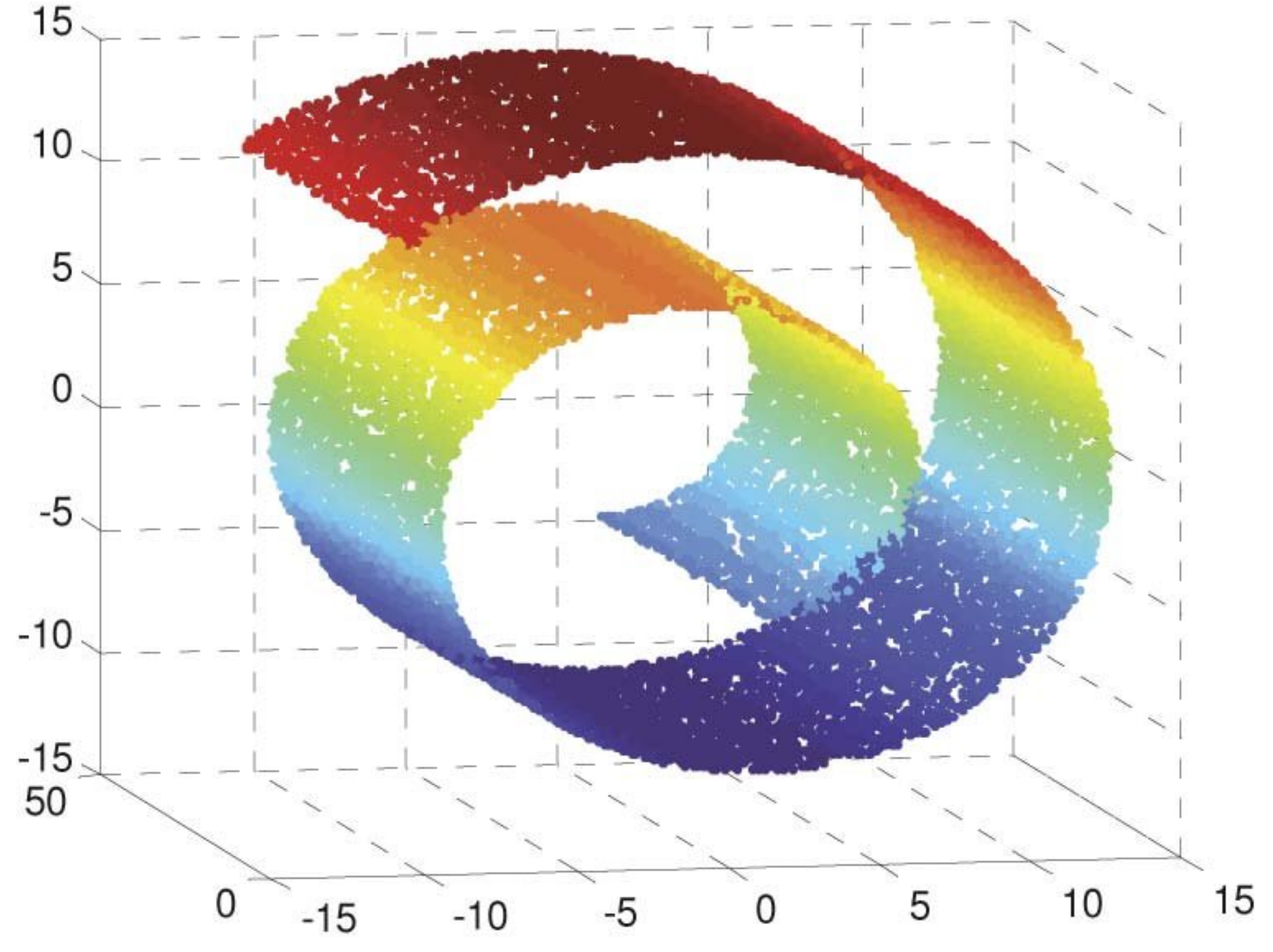
$|y_n - \langle \mathbf{w}, \mathbf{x}_n \rangle - b|^{0.5}$

How to train f

Can our f solve that regression task?



What can we do?



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 $\text{rank}([\mathbf{x}_1, \ldots, \mathbf{x}_N]) \geq N$ then the training loss is 0!

What can we do?

Featurization

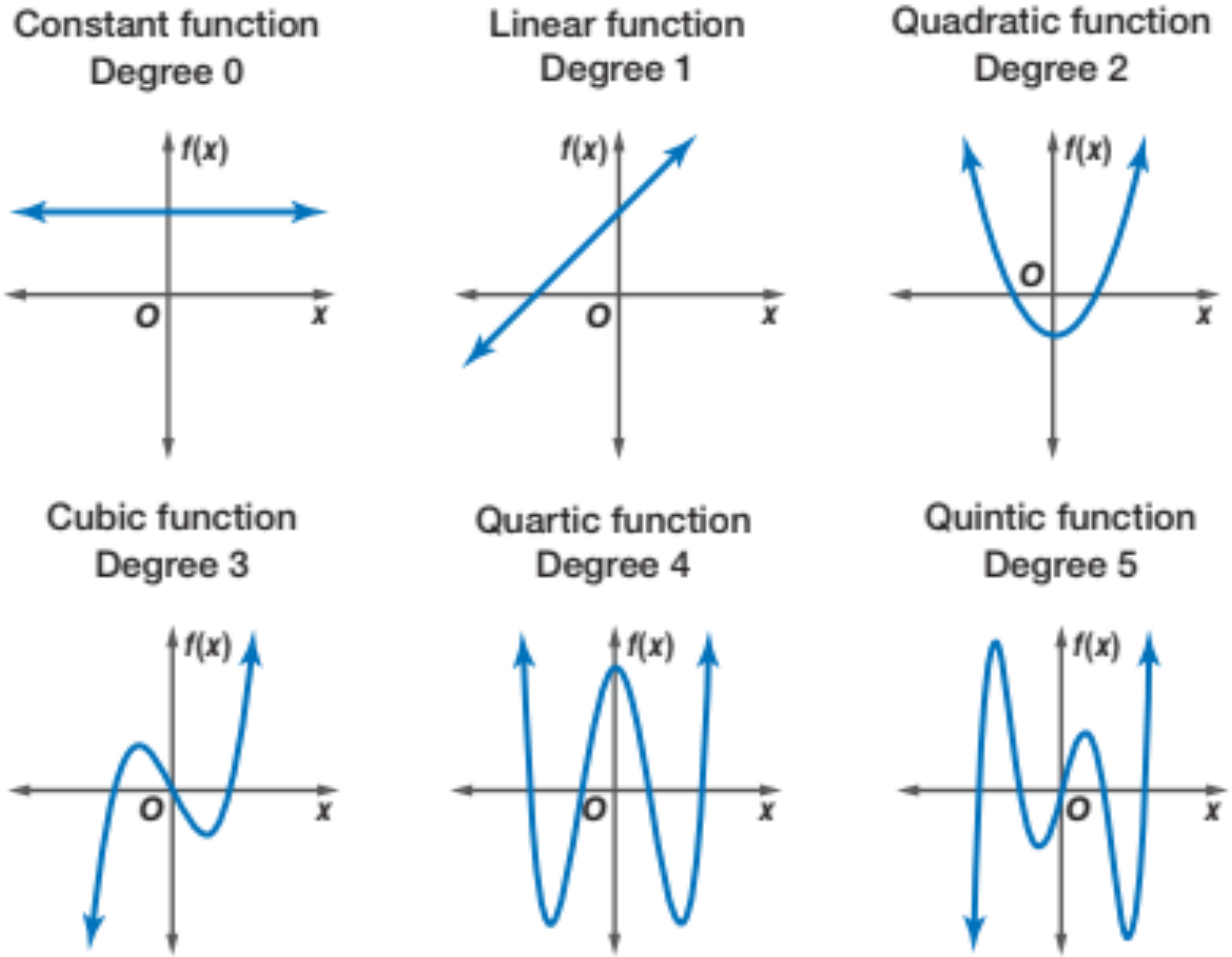
- Add all sort of nonlinear transforms of the data prior to your linear model
- $\mathbf{x}_n \leftarrow (\mathbf{x}_n, \mathbf{x}_n^2, \log(\mathbf{x}_n), \ldots) \in \mathbb{R}^{D'}$
- If rank($[\mathbf{x}_1, \ldots, \mathbf{x}_N]) \geq N$ then the training loss is 0!

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!!!

How to evaluate f

$\mathbb{X}$ $\quad \mathbf{x}_1\ \mathbf{x}_2\ \mathbf{x}_3 \cdots \quad \mathbf{x}_N$

$\mathbb{Y}$ $\quad \mathbf{y}_1\ \mathbf{y}_2\ \mathbf{y}_3 \cdots \quad \mathbf{y}_N$

Training set

Optimize w and b

Valid set

CV p

Test set

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)

Questions?