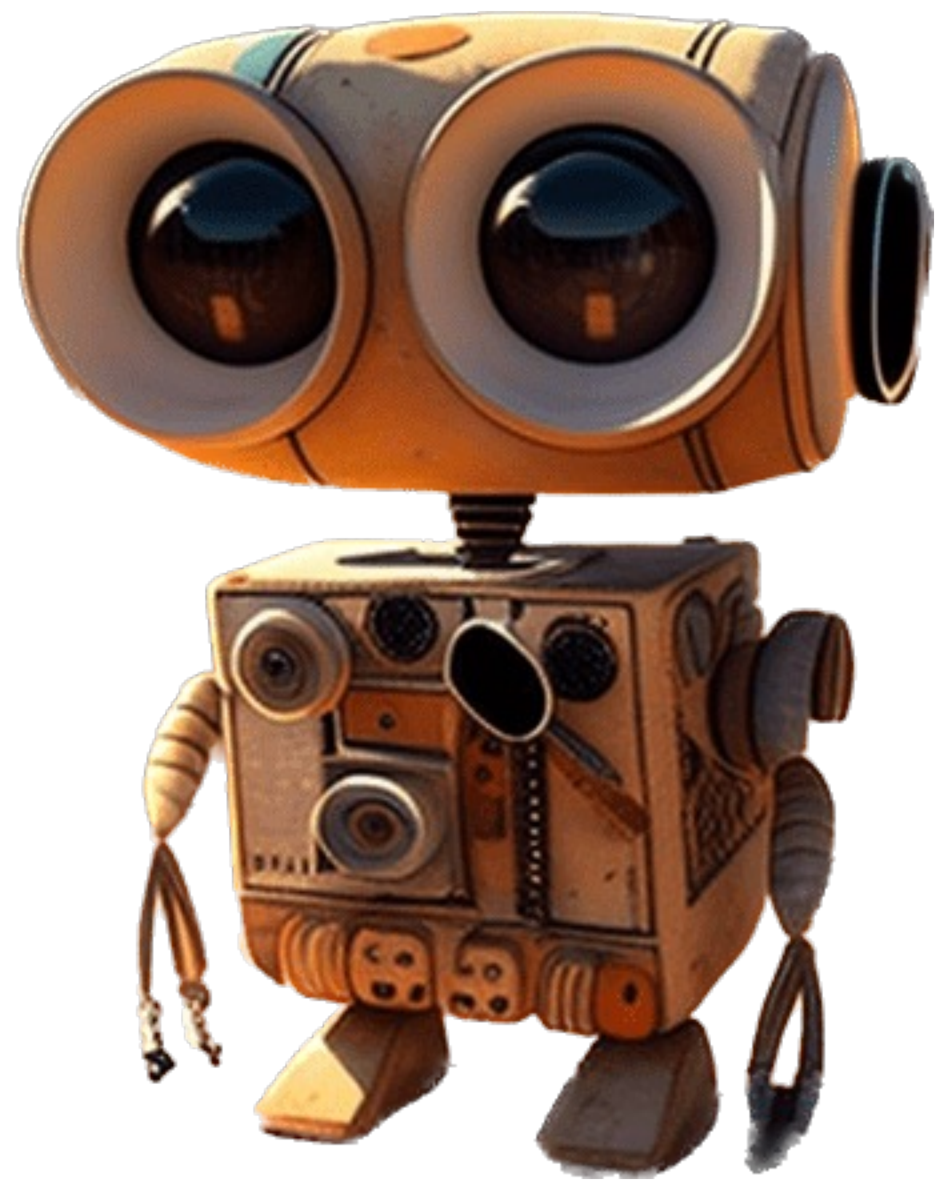
Aprendizaje automático: riesgos computacionales y sociales

Antonio Bahamonde

Universidad de Oviedo
Academia Asturiana de Ciencia e Ingeniería

Introducción



- Aprendizaje automático: un punto de vista actual
- Riesgos
  - Computacionales: generalización, sesgo (bias)-varianza, regularización
  - Sociales: laborales, empresariales, humanos

Contents

- Linear regression
    - Gradient descent
    - Probabilistic interpertation
- Generalization
- Bias and Variance
- Regularization
- Riesgos sociales del AA (LLM)

- Reference (main)
    - Main notes, CS229, Stanford University (Andrew Ng)

Linear regression

Linear regression

Living area (feet^2)	#bedrooms	Price (1000$s)
2104	3	400
1600	3	330
2400	3	369
1416	2	232
3000	4	540
$\vdots$	$\vdots$	$\vdots$

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

θ_i's are the **parameters** (also called **weights**) parameterizing the space of linear functions mappings

$$h : \mathcal{X} \mapsto \mathcal{Y}$$

Linear regression

When there is no risk of confusion, we will drop the θ subscript in

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

write it more simply as h(x). To simplify our notation, we also introduce the convention of letting $x_0 = 1$ (this is the **intercept** term), so that

$$h(x) = \sum_{i=0}^{d} \theta_i x_i = \theta^T x,$$

Linear regression

Given a training set, how do we pick, or learn, the parameters θ?

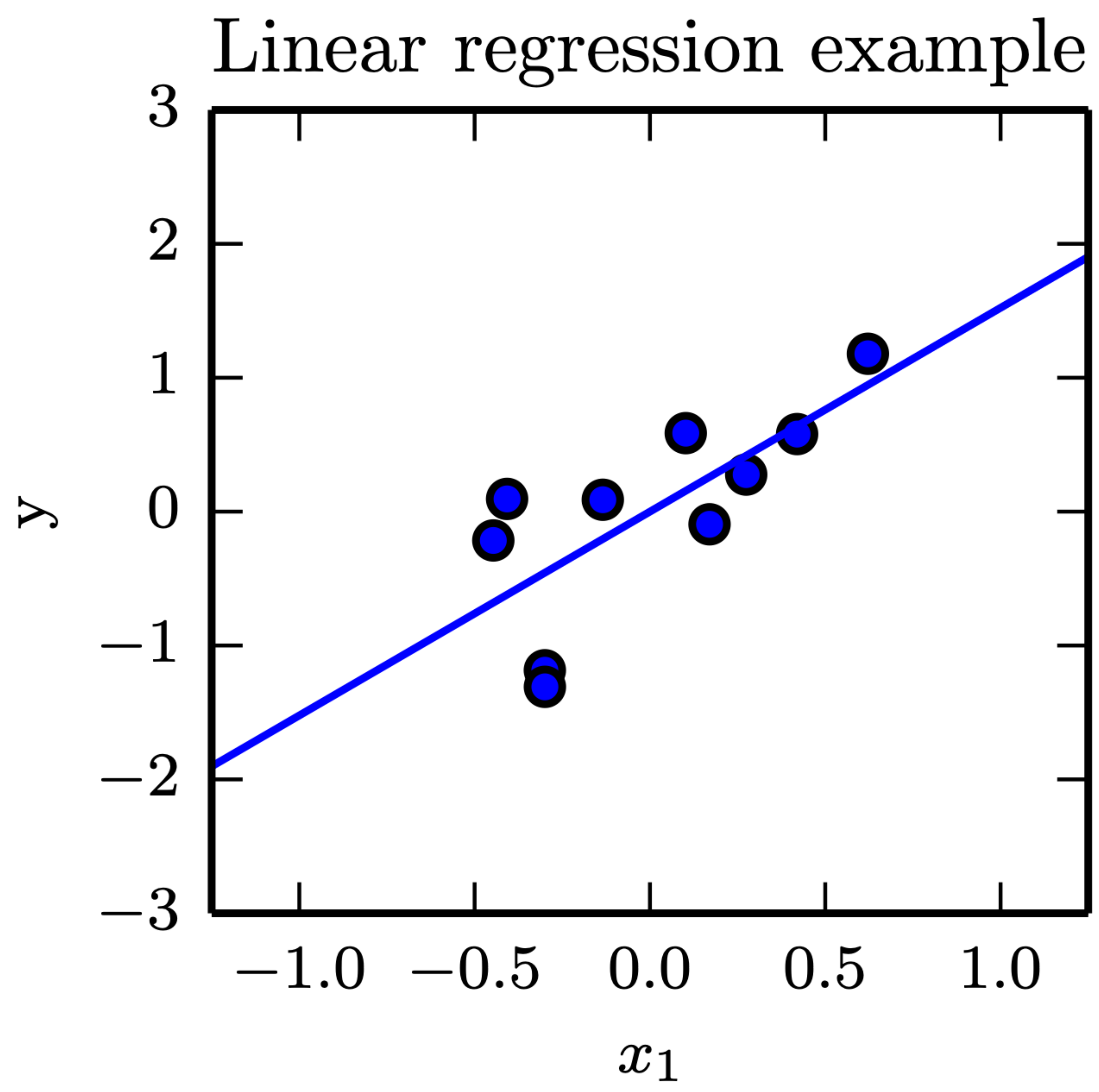
One reasonable method seems to be to make h(x) close to y, at least for the training examples we have.

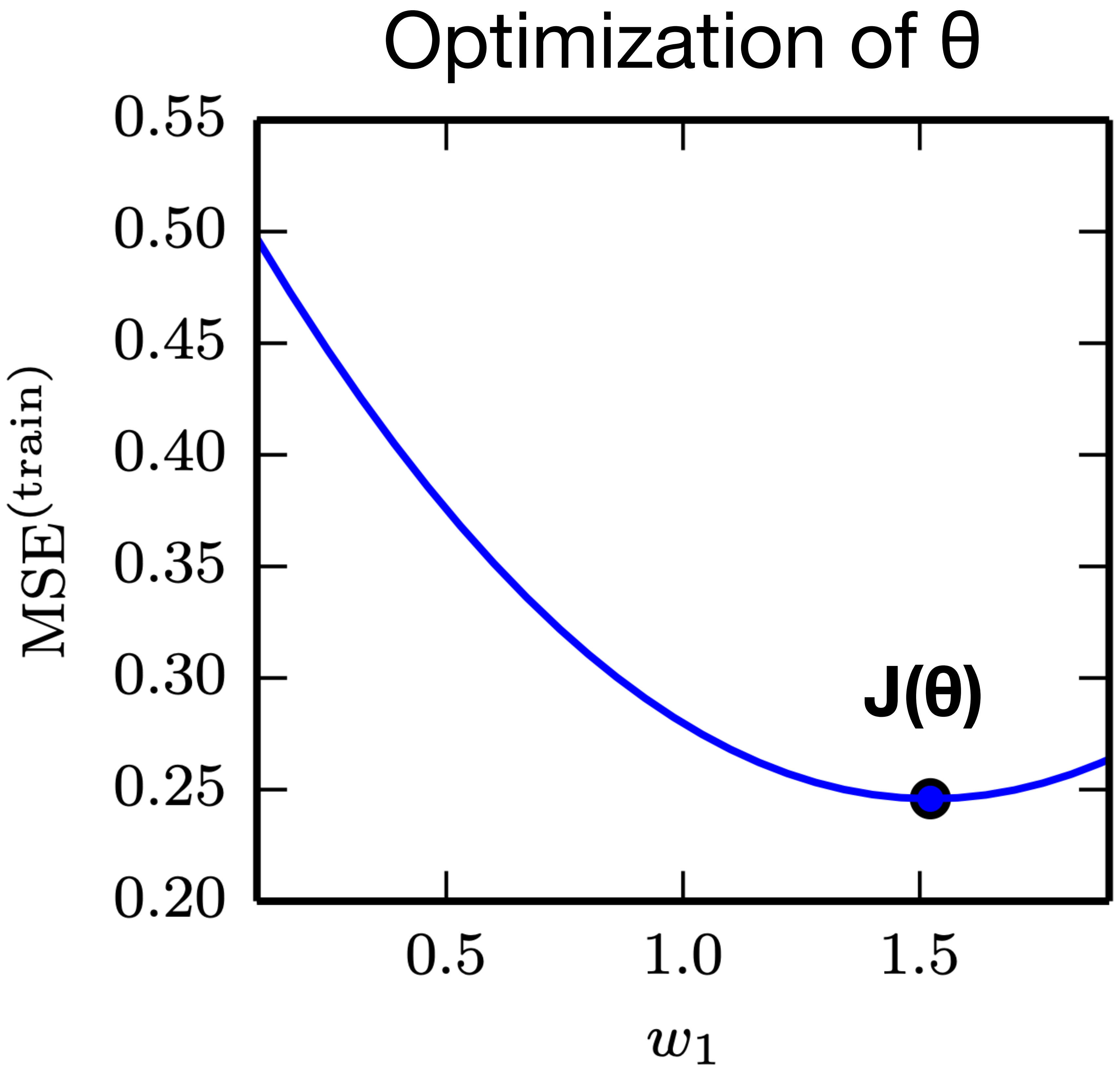
To formalize this, we will define a function that measures, for each value of the θ's, how close the h(x)'s are to the corresponding y's . We define the **cost function**:

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{n}(h_\theta(x^{(i)}) - y^{(i)})^2.$$

least-squares cost function that gives rise to the ordinary least squares regression model

Linear Regression





$y \approx \theta_1 x_1$

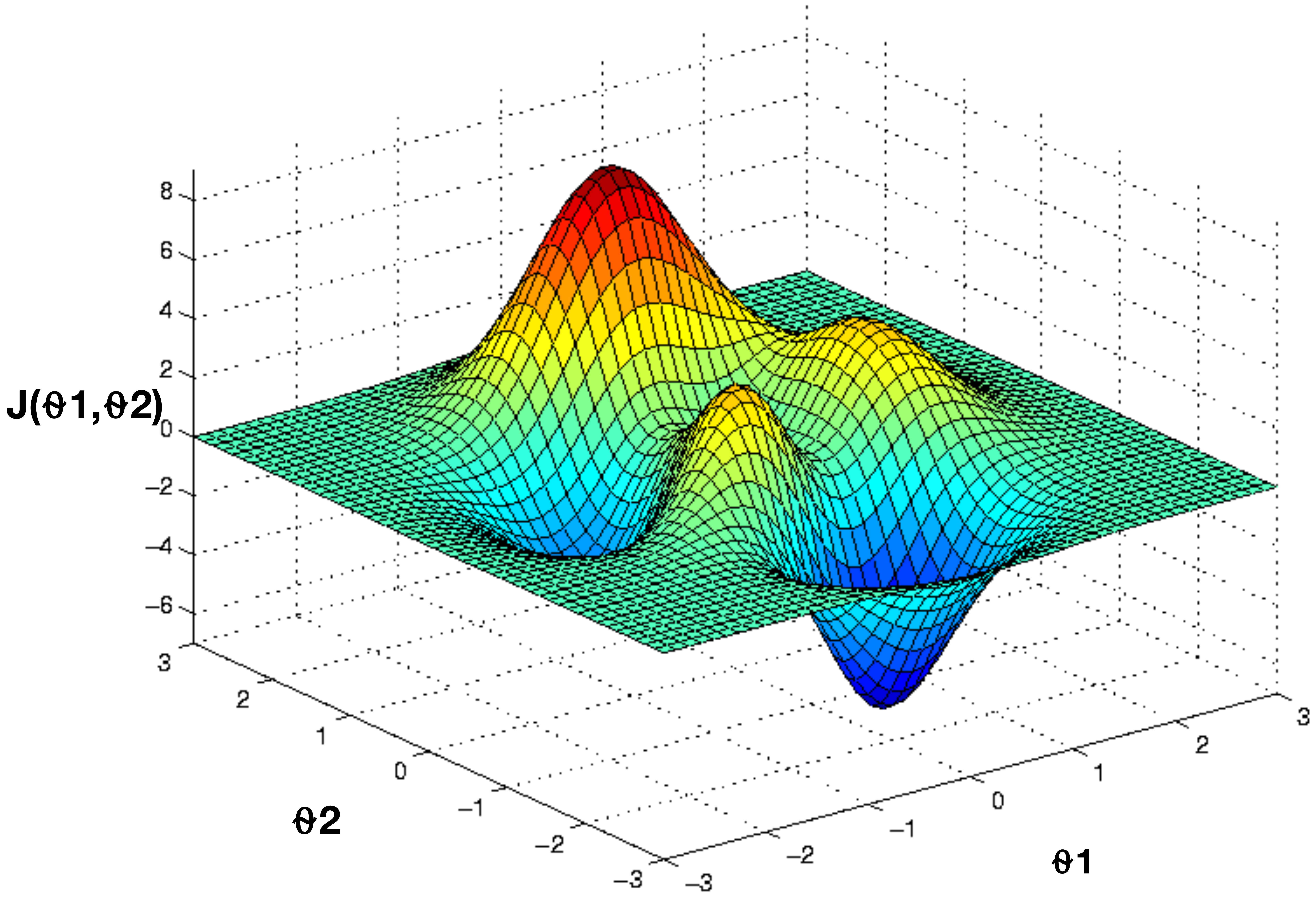
Gradient descent

Gradient descent

Given a function J(ϑ1,ϑ2) we look for the values of ϑ1 and ϑ2 that make the value of J to a minimum

Algorithm:

- Assign arbitrary values to ϑ1 and ϑ2
- Modify the values of ϑ1 and ϑ2 to reduce the value of J
  Until we believe that we have reached a minimum


J(θ1,θ2)
8
6
4
2
0
−2
−4
−6
3
2
1
0
−1
−2
−3
θ2
−3
−2
−1
0
1
2
3
θ1

Gradient descent

To solve

$$\theta \leftarrow \underset{\theta}{\operatorname{argmin}}\ \mathbf{J}(\theta)$$

GD is the algorithm

$\theta \longleftarrow \theta_0$

repeat

$$\theta \longleftarrow \theta - \gamma * \frac{\partial J}{\partial \theta}$$

until no more improvement can be reached

return θ

Linear regression

$\gamma \dots \alpha$

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta).$$

$$\boxed{J(\theta) = \frac{1}{2} \sum_{i=1}^{n} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2.}$$

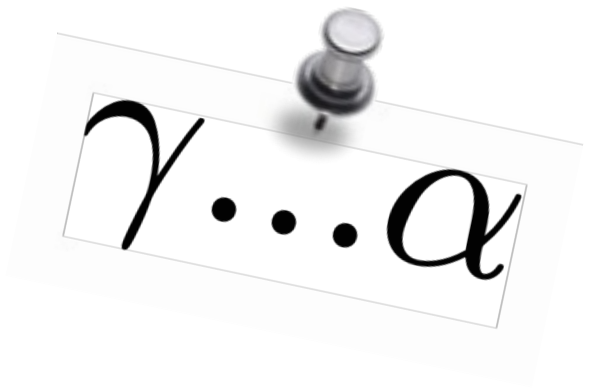
$$\begin{aligned}
\frac{\partial}{\partial \theta_j} J(\theta) &= \frac{\partial}{\partial \theta_j} \frac{1}{2} \left(h_\theta(x) - y\right)^2 \\
&= 2 \cdot \frac{1}{2} \left(h_\theta(x) - y\right) \cdot \frac{\partial}{\partial \theta_j} \left(h_\theta(x) - y\right) \\
&= \left(h_\theta(x) - y\right) \cdot \frac{\partial}{\partial \theta_j} \left(\sum_{i=0}^{d} \theta_i x_i - y \right) \\
&= \left(h_\theta(x) - y\right) x_j
\end{aligned}$$

$$\theta_j := \theta_j + \alpha \left(y^{(i)} - h_\theta(x^{(i)}) \right) x_j^{(i)}.$$

Linear regression

For a single training example, this gives the LMS (“least mean squares”) update rule

$$\theta_j := \theta_j + \alpha \left(y^{(i)} - h_\theta(x^{(i)})\right) x_j^{(i)}.$$



the magnitude of the update is proportional to the error term $(y^{(i)} - h_\theta(x^{(i)}))$:

- if we are encountering a training example on which our prediction nearly matches the actual value of $y^{(i)}$, then we find that there is little need to change the parameters;
- a larger change to the parameters will be made if our prediction has a large error

Probabilistic interpretation

$$y^{(i)} = \theta^T x^{(i)} + \epsilon^{(i)},$$

$$p(\epsilon^{(i)}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\epsilon^{(i)})^2}{2\sigma^2}\right)$$

$$p(y^{(i)}|x^{(i)};\theta) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right).$$

$$\begin{aligned} L(\theta) &= \prod_{i=1}^{n} p(y^{(i)} \mid x^{(i)};\theta) \\ &= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right) \end{aligned}$$

Probabilistic interpretation

$$\begin{aligned}
\ell(\theta) &= \log L(\theta) \\
&= \log \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right) \\
&= \sum_{i=1}^{n} \log \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right) \\
&= n \log \frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{\sigma^2} \cdot \frac{1}{2} \sum_{i=1}^{n} (y^{(i)} - \theta^T x^{(i)})^2.
\end{aligned}$$

Probabilistic interpretation

Least-squares regression corresponds to finding the maximum likelihood estimate of θ.

θ did not depend on what was σ, even if σwere unknown.

$$\frac{1}{2}\sum_{i=1}^{n}(y^{(i)} - \theta^T x^{(i)})^2,$$

Generalization

Generalization

- The **generalization** of machine learning models: their performances on unseen test example
- Supervised learning problems, given a training dataset, we typically learn a model h_θ by minimizing a loss/cost function $J(\theta)$, which encourages h_θ to fit the data. This loss function for training purposes is oftentimes referred to as the **training loss/error/cost**.

Generalization

- Minimizing the training loss is not our ultimate goal—it is merely our approach towards the goal of learning a predictive model. The most important evaluation metric of a model is the loss on unseen test examples, which is oftentimes referred to as the **test error**.
    - minimizing the training error may not always lead to a small test error
    - the model **overfits** the data if the model predicts accurately on the training dataset but doesn't generalize well to other test examples, that is, if the training error is small but the test error is large.

Generalization

Test distribution D.
The expected loss/error over the randomness of the test example is called the **test loss/error**

$$L(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[(y - h_\theta(x))^2]$$

We'll see how the test error is influenced by the learning procedure, especially the choice of model parameterizations.

We will decompose the test error into "bias" and "variance" and their tradeoffs.

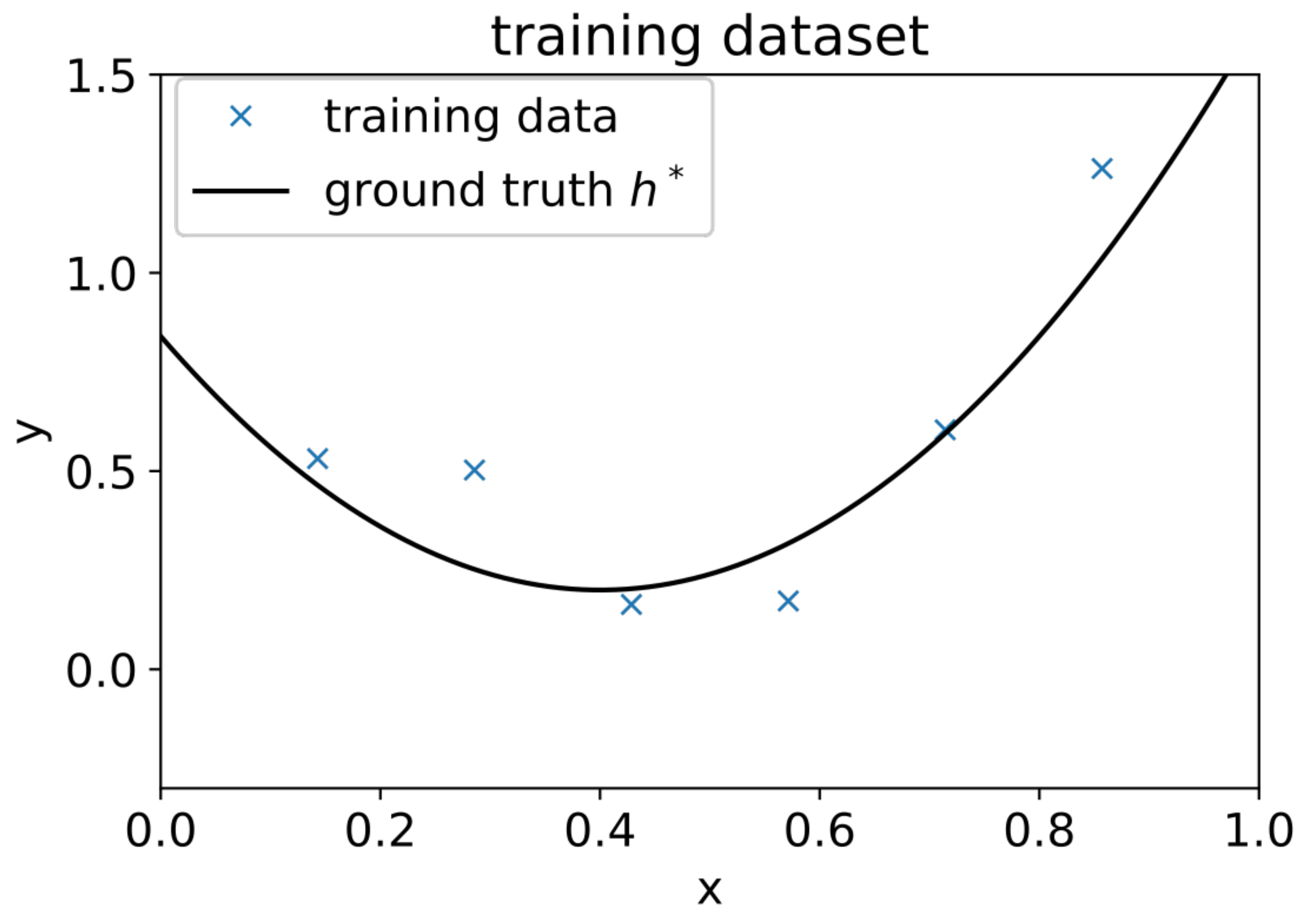
We will discuss when overfitting and underfitting will occur and be avoided.

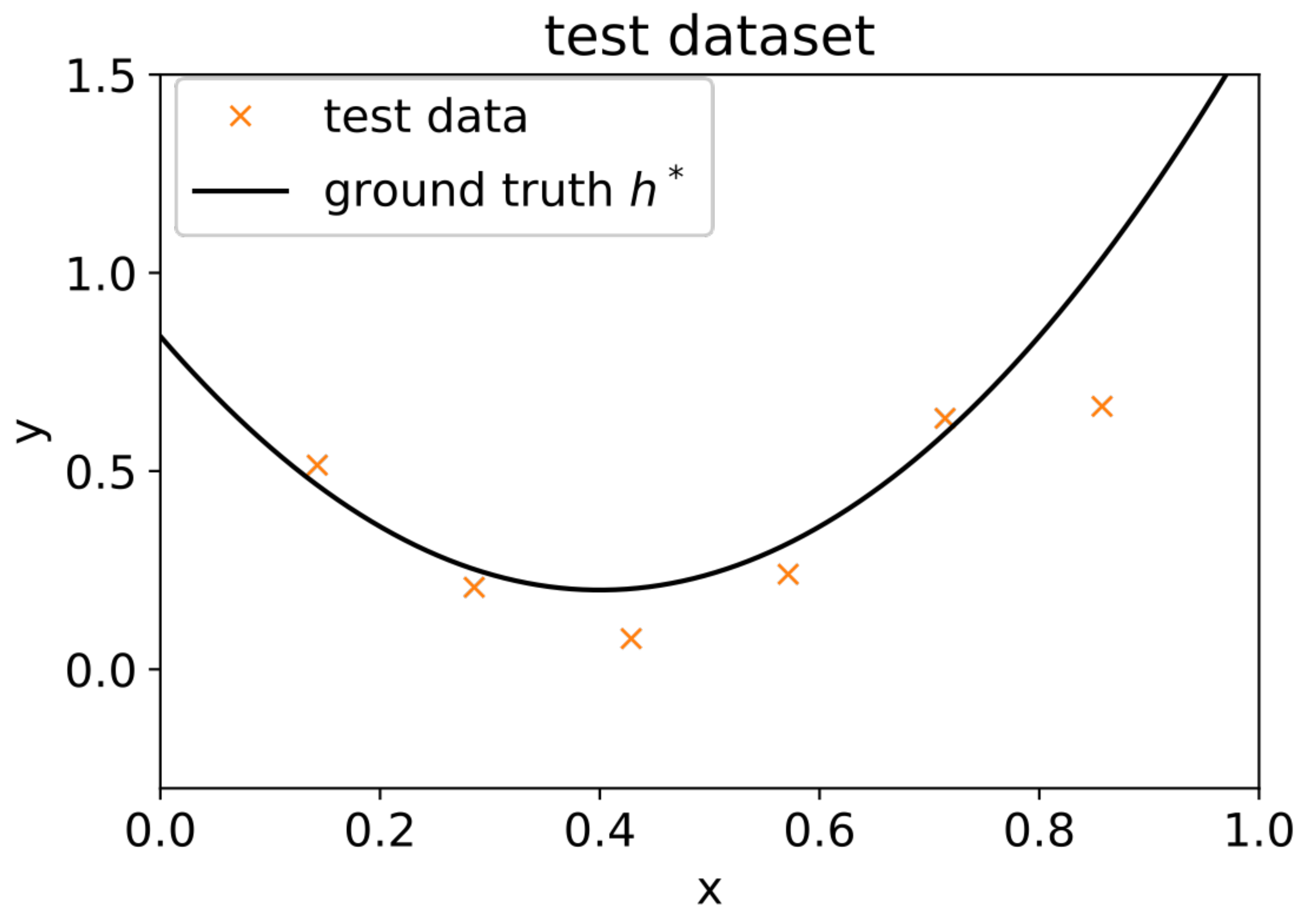
Bias and variance

Bias and variance

- The training inputs $x^{(i)}$'s are randomly chosen and the outputs $y^{(i)}$ are generated by $y^{(i)} = h^*(x^{(i)}) + \xi^{(i)}$ where the function $h^*(\cdot)$ is a quadratic function $\xi^{(i)}$ is the observation noise assumed to be generated from $\sim N(0,\sigma^2)$.

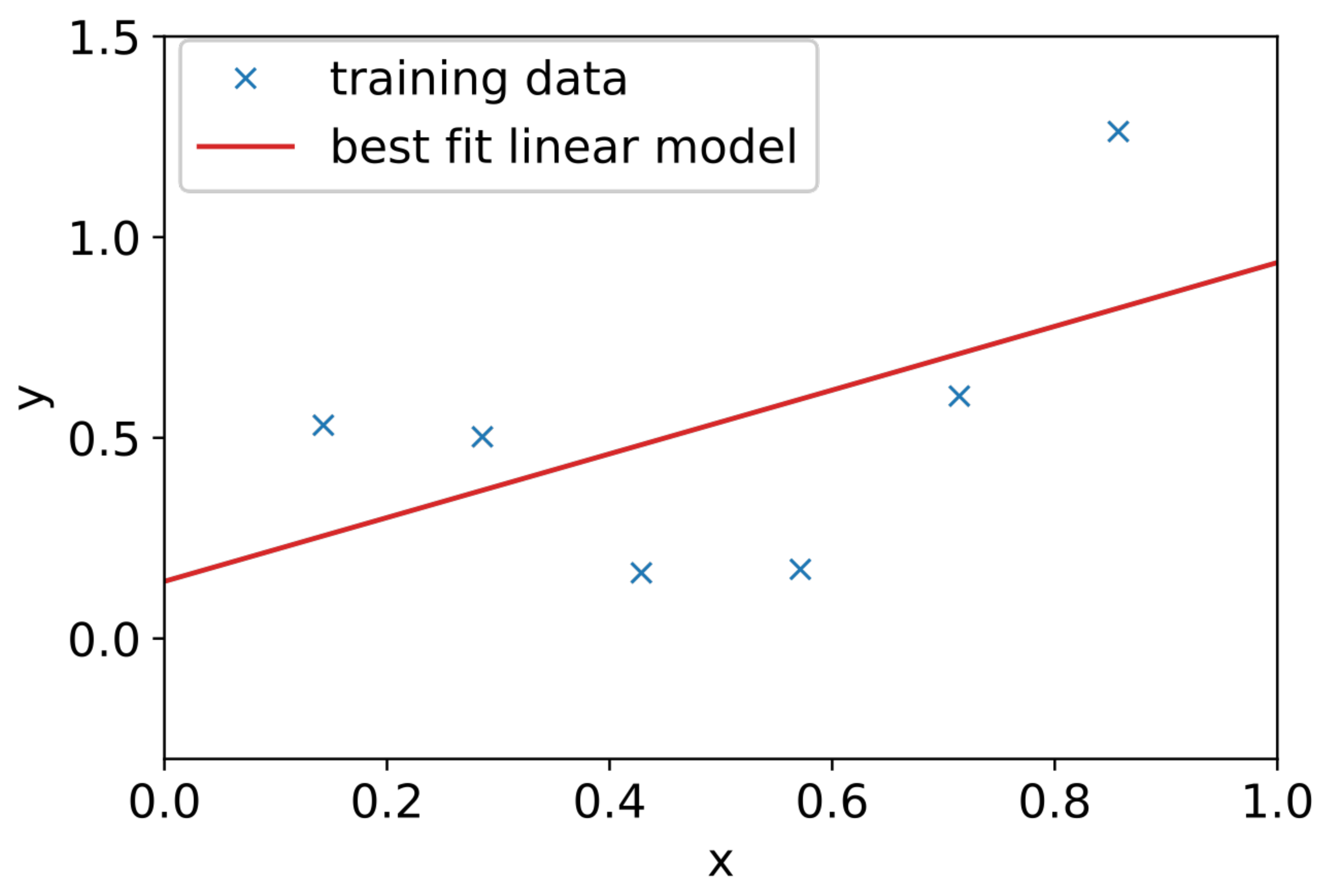
- A test example (x,y) also has the same input-output relationship $y = h^*(x) + \xi$ where $\xi \sim N(0,\sigma^2)$. It's impossible to predict the noise ξ, and therefore essentially our goal is to recover the function $h^*(\cdot)$

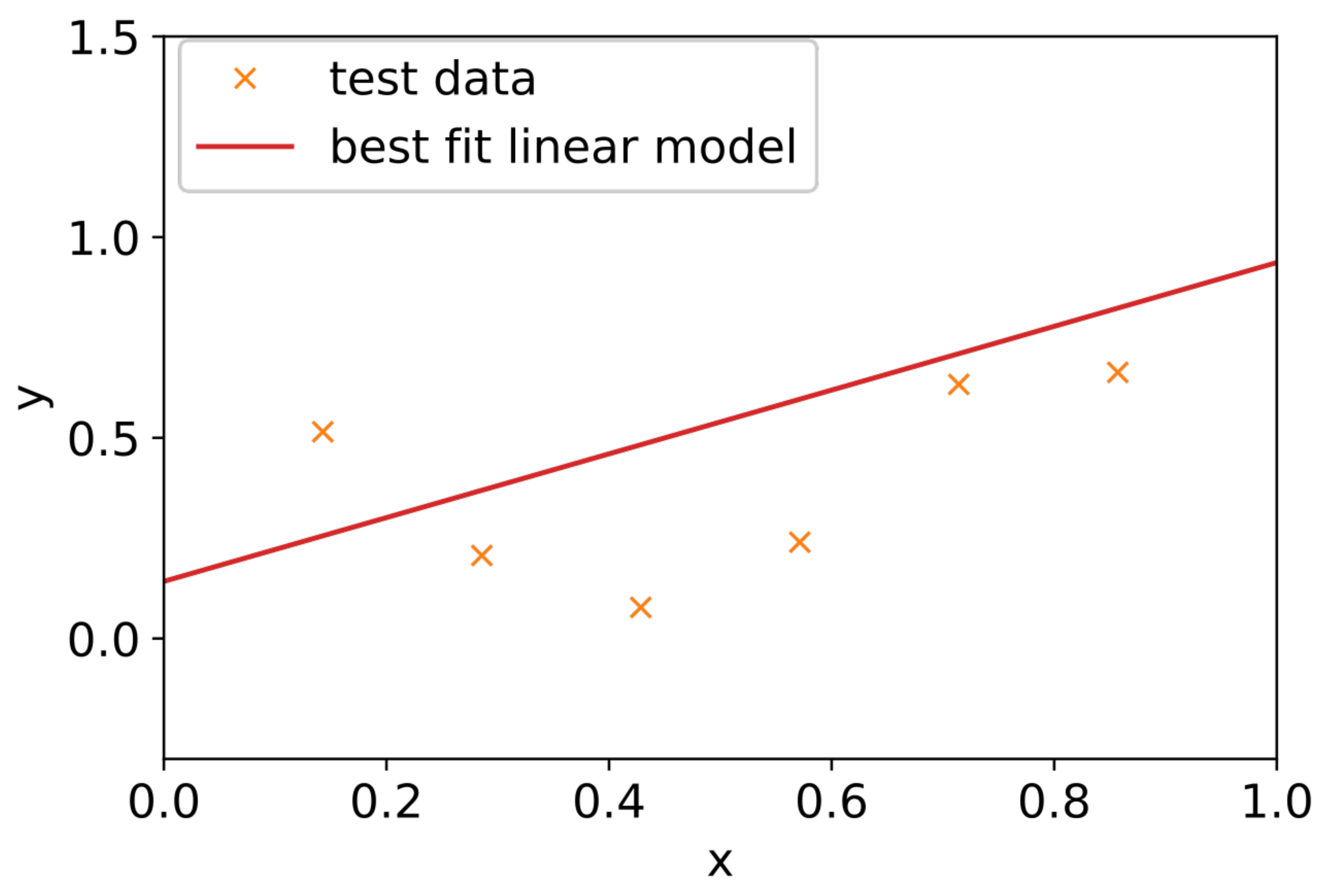




Bias and variance

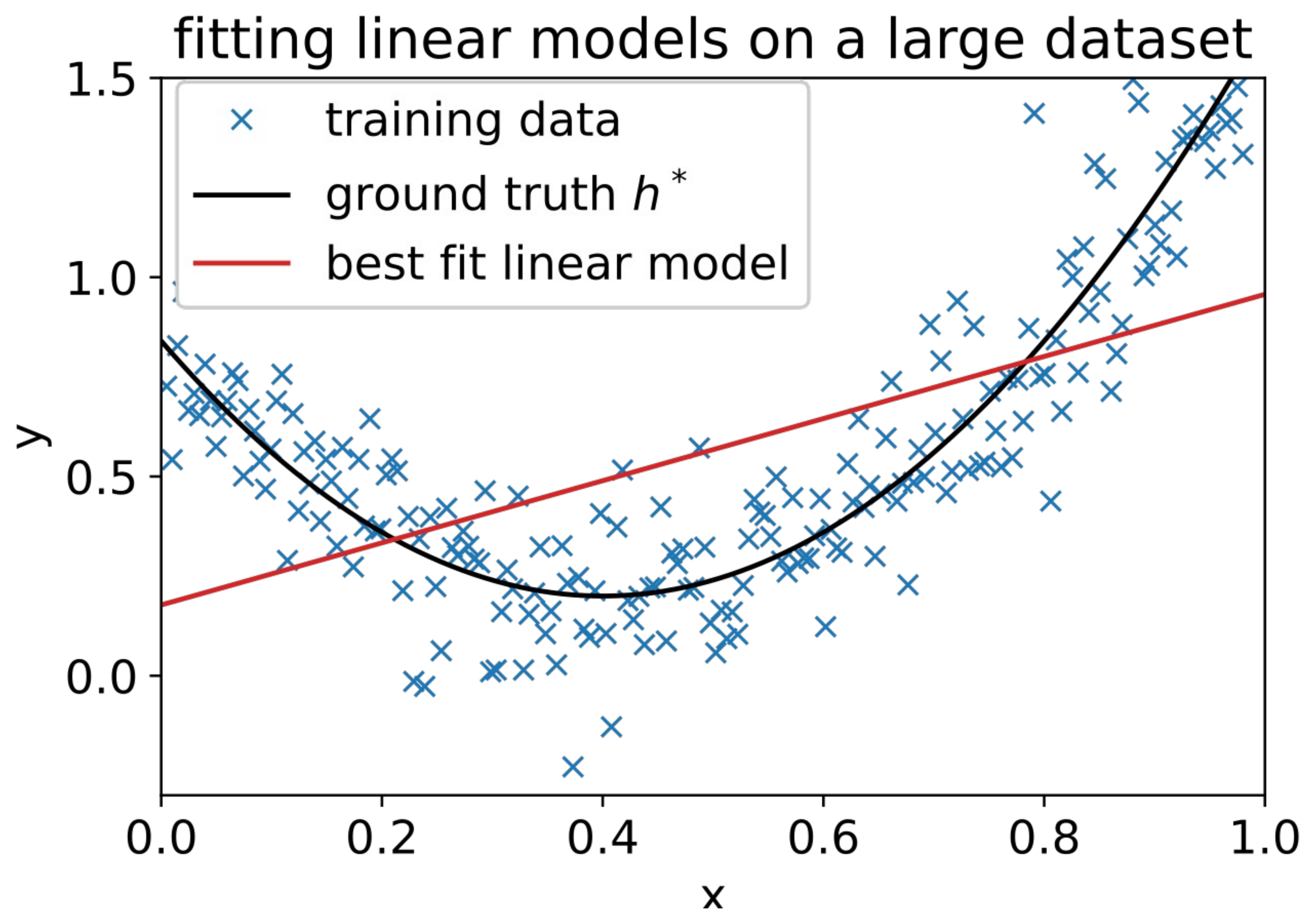
The best fit linear model has large training and test errors

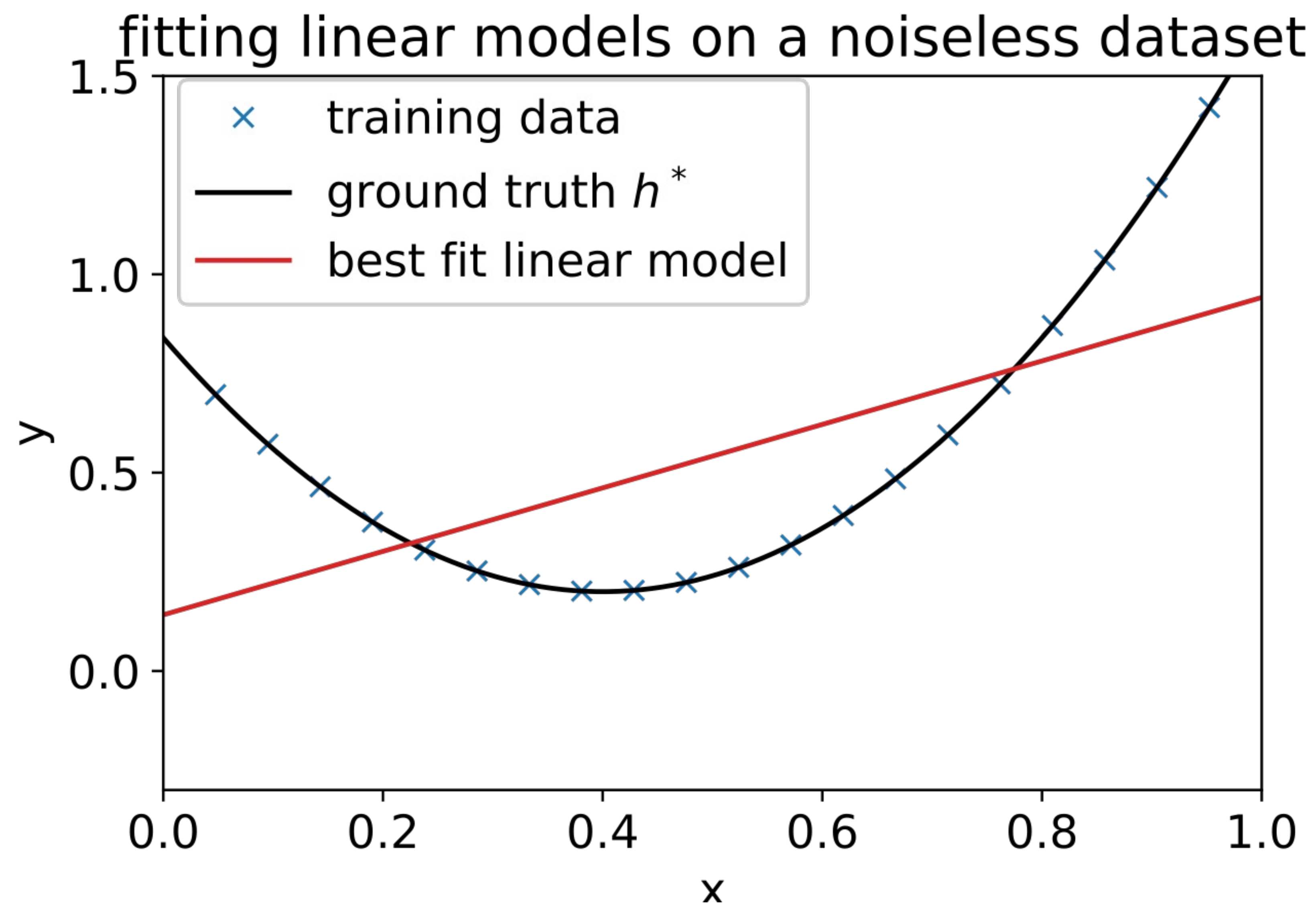




Bias and variance

The issue cannot be mitigated with more training examples—even with a very large amount of, or even infinite training examples, the best fitted linear model is still inaccurate and fails to capture the structure of the data. Even if the noise is not present in the training data, the issue still occurs





Bias and variance

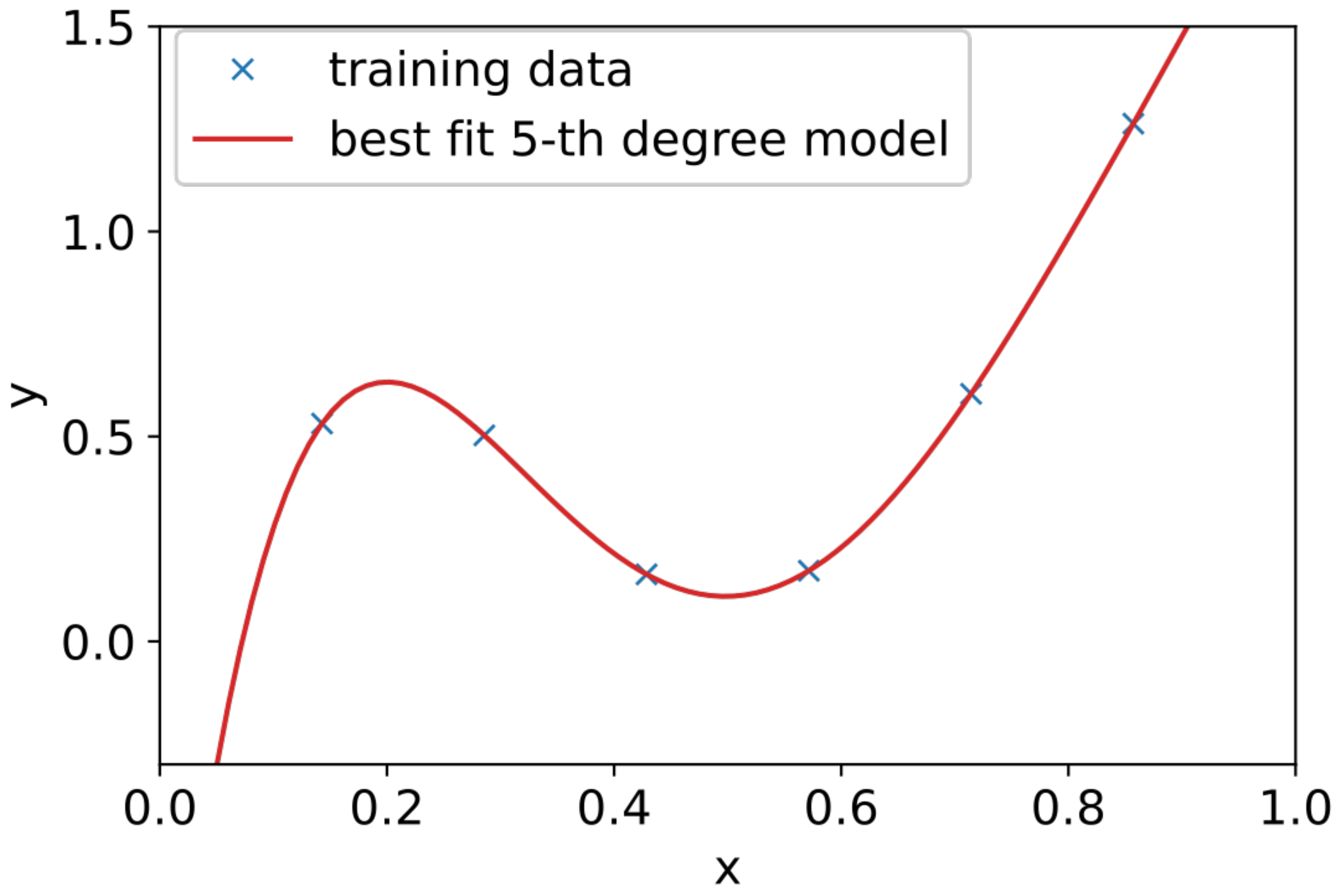
The linear model family's inability to **capture the structure** in the data—linear models cannot represent the true quadratic function h*—, but not the lack of the data.

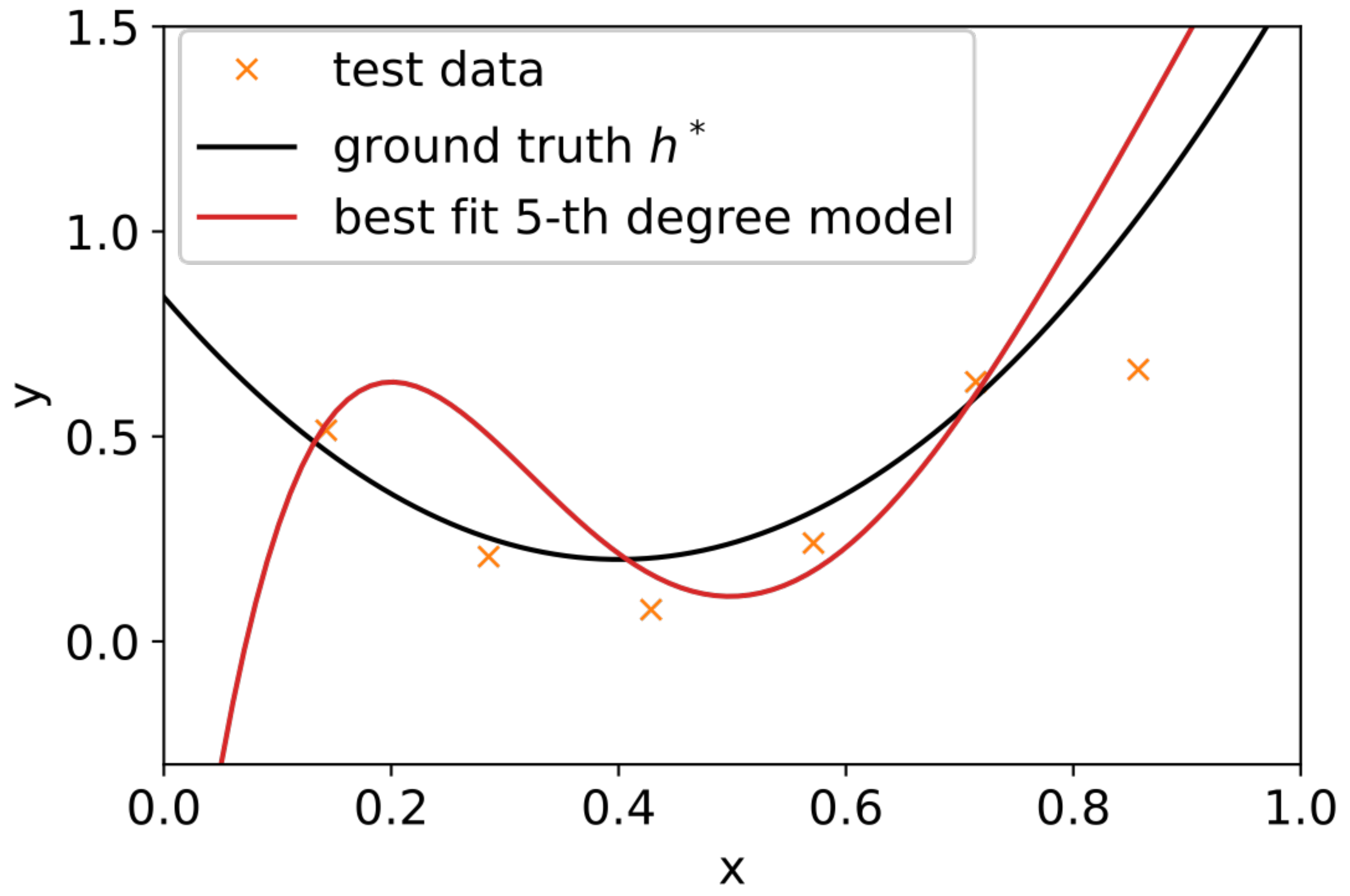
Informally, we define the **bias** of a model to be the test error even if we were to fit it to a very (say, infinitely) large training dataset.

The linear model suffers from large **bias**, and underfits the data.

Bias and variance

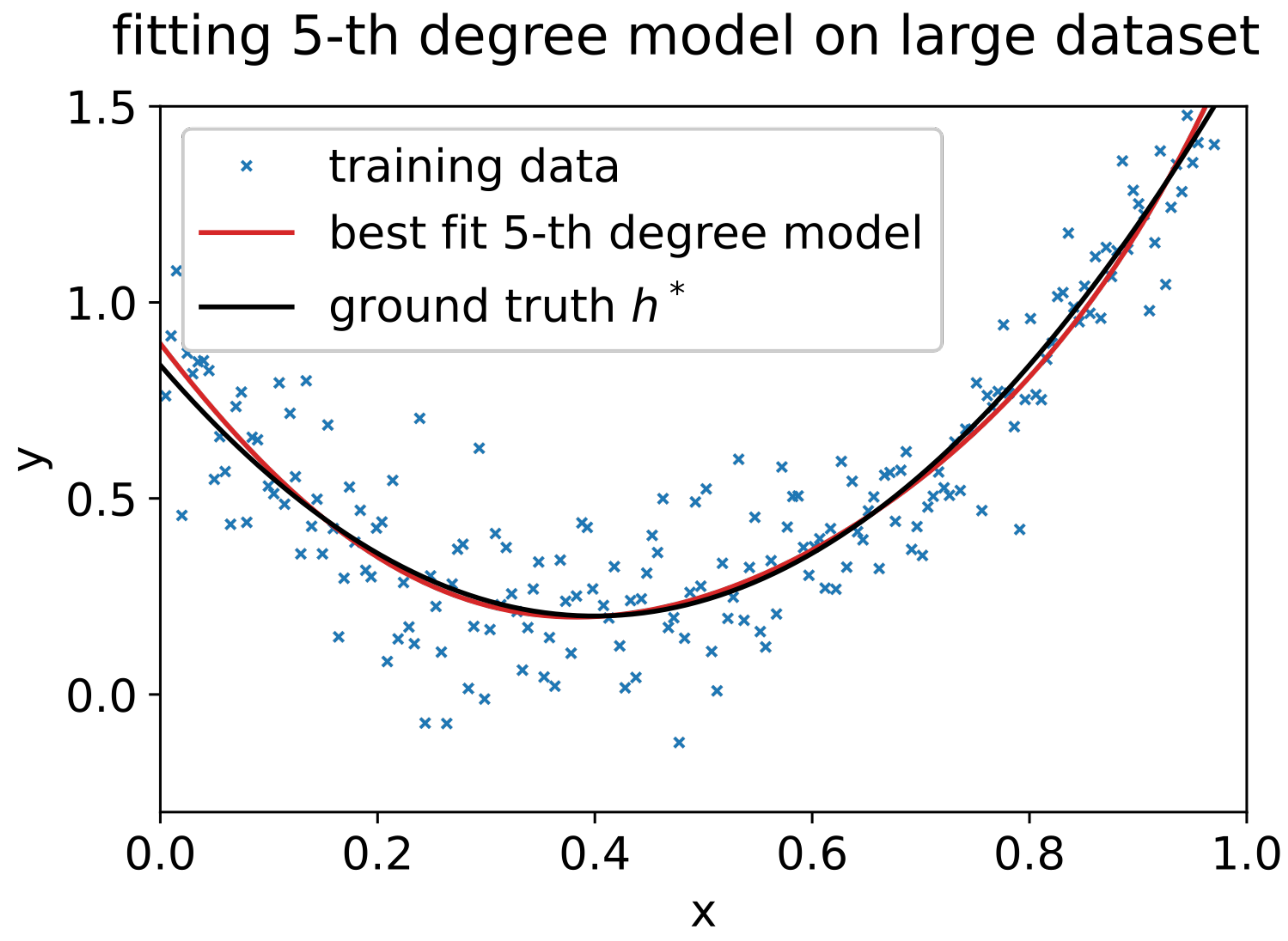
Next, we fit a 5th-degree polynomial to the data.





The model learnt from the training set does **not generalize** well to other test examples

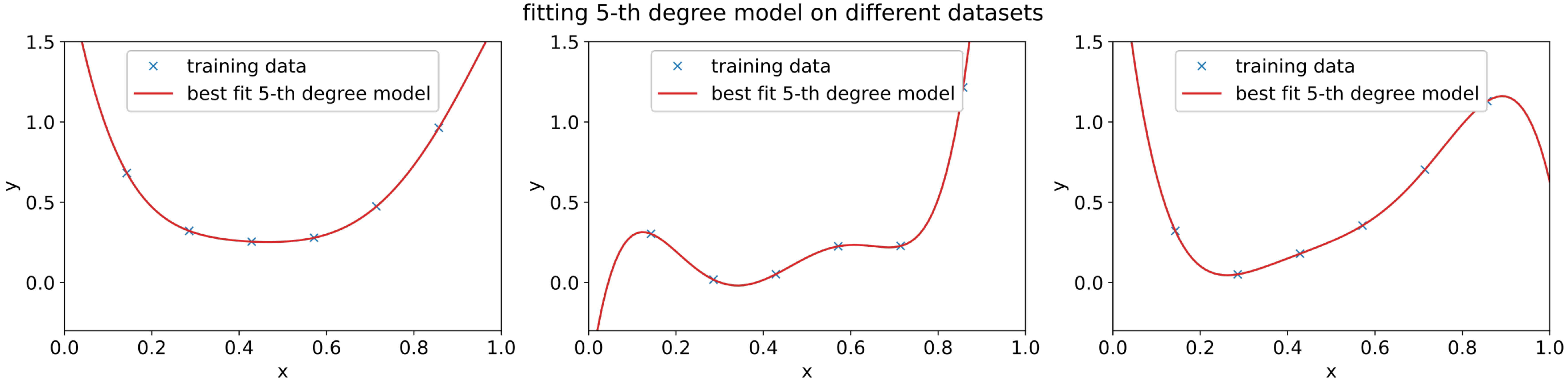
Bias and variance



The **bias** of the 5-th degree polynomials **is small**—if we were to fit to an extremely large dataset, the resulting model would be close to a quadratic function and be accurate (because the family of 5-th degree polynomials contains all the quadratic functions)

Bias and variance

The failure of fitting 5-th degree polynomials can be captured by another component of the test error, called **variance** of a model fitting procedure



there is a large risk that we're fitting patterns in the data that happened to be present in our small, finite training set, but that do not reflect the wider pattern of the relationship between x and y. These "**spurious**" patterns in the training set are (mostly) due to the observation noise ξ(i), and fitting these spurious patters results in a model with large test error.

Bias and variance

The **variance** can be intuitively characterized by the amount of variations across models learnt on multiple different training datasets (drawn from the same underlying distribution).

Bias and variance tradeoff

If our model is **too “simple”** (has very few parameters), then it may have large bias (but small variance), and it typically may suffer from **underfittng**.

If it is **too “complex”** and has very many parameters, then it may suffer from large variance (but have smaller bias), and thus **overfitting**.

A mathematical decomposition of error (for regression)

$$\begin{aligned}\mathrm{MSE}(x) &= \sigma^2 + \mathbb{E}[(h^\star(x) - h_S(x))^2] \\ &= \sigma^2 + (h^\star(x) - h_{\text{avg}}(x))^2 + \mathbb{E}[(h_{\text{avg}} - h_S(x))^2] \\ &= \underbrace{\sigma^2}_{\text{unavoidable}} + \underbrace{(h^\star(x) - h_{\text{avg}}(x))^2}_{\triangleq\ \text{bias}^2} + \underbrace{\mathrm{var}(h_S(x))}_{\triangleq\ \text{variance}}\end{aligned}$$

- h_{avg} can be thought of as the best possible model learned even with infinite data
- The bias captures the part of the error that are introduced due to the lack of expressivity of the model. It is not due to the lack of data

Regularization

Regularization

- **Overfitting** is typically a result of using too complex models, and we need to choose a proper model complexity to achieve the optimal bias-variance tradeoff

- When the **model complexity** is measured by the number of parameters, we can vary the size of the model (e.g., the number of features)

- However, the correct, informative complexity measure of the models can be a function of the **parameters** (e.g., ℓ_2 norm of the parameters), which may not necessarily depend on the number of parameters. Then we use regularization

- **Regularization** is an important technique in machine learning that controls the model complexity and prevents overfitting

Regularization

- Regularization typically involves adding an additional term, called a **regularizer** and denoted by $R(\theta)$ here, to the training loss/cost function:

$$J_\lambda(\theta) = J(\theta) + \lambda R(\theta)$$

- J_λ is often called the **regularized loss**, and $\lambda \geq 0$ is called the **regularization parameter**.

- The regularizer $R(\theta)$ is typically chosen to be some measure of the complexity of the model θ. Thus, when using the regularized loss, we aim to find a model that both fit the data (a small loss $J(\theta)$) and have a small model complexity (a small $R(\theta)$).

- When λ is a sufficiently small positive number, minimizing the regularized loss is effectively minimizing the original loss with the regularizer as the tie-breaker. When the regularizer is extremely large, then the original loss is not effective (and likely the model will have a large bias.)

Regularization

- The most commonly used regularization is perhaps ℓ_2 regularization, where
$$R(\theta) = \frac{1}{2}\|\theta\|_2^2$$

- It encourages the optimizer to find a model with small ℓ_2 norm. In deep learning, it's oftentimes referred to as weight decay, because gradient descent with learning rate η on the regularized loss $R_\lambda(\theta)$ is equivalent to shrinking/decaying θ by a scalar factor of 1 – ηλ and then applying the standard gradient

$$\theta \leftarrow \theta - \eta \nabla J_\lambda(\theta) = \theta - \eta\lambda\theta - \eta\nabla J(\theta)$$
$$= \underbrace{(1-\lambda\eta)\theta}_{\text{decaying weights}} - \eta\nabla J(\theta)$$

Regularization (sparsity)

- Besides encouraging simpler models, regularization can also impose **inductive biases** or structures on the model parameters. For example, suppose we had a prior belief that the number of non-zeros in the ground-truth model parameters is small,—which is oftentimes called **sparsity** of the model—, we can impose a regularization on the number of non-zeros in θ

- The sparsity of the parameters is not a continuous function of the parameters, and thus we **cannot** optimize it with (stochastic) gradient descent. A common relaxation is to use as a continuous surrogate the norm-1 (also called **LASSO**)

$$R(\theta) = ||\theta||_1$$

- Norms 1 and 2 are the most commonly used regularizers for **linear models**. In deep learning norm 2

Riesgos sociales del AA (LLM)

GPTs are GPTs: An Early Look at the Labor Market Impact Potential of Large Language Models

Tyna Eloundou[1], Sam Manning[1,2], Pamela Mishkin*[1], and Daniel Rock[3]

[1]OpenAI
[2]OpenResearch
[3]University of Pennsylvania

March 20, 2023

- roles heavily reliant on science and critical thinking skills show a negative correlation with exposure, while programming and writing skills are positively associated with LLM exposure.

- information processing industries (4-digit NAICS) exhibit high exposure, while manufacturing, agriculture, and mining demonstrate lower exposure.

El trabajo requerirá al menos un 50% menos de tiempo

Job Zone	Preparation Required	Education Required	Example Occupations	Median Income	Tot Emp (000)	H α	M α
1	None or little (0-3 months)	High school diploma or GED (optional)	Food preparation workers, dishwashers, floor sanders	$30,230	13,100	3.71	3.84
2	Some (3-12 months)	High school diploma	Orderlies, customer service representatives, tellers	$38,215	73,962	7.03	11.88
3	Medium (1-2 years)	Vocational school, on-the-job training, or associate's degree	Electricians, barbers, medical assistants	54,815	37,881	11.28	13.72
4	Considerable (2-4 years)	Bachelor's degree	Database administrators, graphic designers, cost estimators	$77,345	56,833	22.68	17.82
5	Extensive (4+ years)	Master's degree or higher	Pharmacists, lawyers, astronomers	$81,980	21,221	22.81	13.36

Table 6: Exposure to GPTs by Job Zone

Repercusiones IA

- Disminuye la **Intermediación**

- Organizaciones (debido a la IA) transforman en **plataformas** gestionas por algoritmos que mueven una enorme cantidad de datos. Por ejemplo los bancos, operadoras móviles, eléctricas, seguros …

- Consumo **energético**

¿Por dónde puede ir la solución?

- Humanidad: **creatividad** extra
- Laboral. Avanzar en
  - Intermediación
  - Automatización (plataformas)
  - **Optimización** de
    - procesos más sostenibles
    - la vida (cuidados)

¿Y la universidad?

- Cómo enseñamos?
- Para qué profesiones preparamos a nuestros estudiantes?
- Qué destrezas deberán tener?

¿Y ahora qué?

- Humildad género humano
- Humildad metodológica

¿Y ahora qué?

Andrew Ng
@AndrewYNg

When I think of existential risks to large parts of humanity:
* The next pandemic
* Climate change→massive depopulation
* Another asteroid
AI will be a key part of our solution. So if you want humanity to survive & thrive the next 1000 years, lets make AI go faster, not slower.

6:33 PM · May 30, 2023 · **1.1M** Views

824 Retweets **175** Quotes **4,110** Likes **205** Bookmarks

¿Y ahora qué?

Yann LeCun
@ylecun

Super-human AI is nowhere near the top of the list of existential risks.
In large part because it doesn't exist yet.

Until we have a basic design for even dog-level AI (let alone human level), discussing how to make it safe is premature.


Muchas
gracias