Machine Learning

Angshuman Paul

Assistant Professor

Department of Computer Science & Engineering

Syllabus

- **Fractal I: Supervised Learning**
- Introduction: Definitions, Datasets for Machine Learning, Different Paradigms of Machine Learning, Data Normalization, Hypothesis Evaluation, VC-Dimensions and Distribution, Bias-Variance Tradeoff, Regression
- Bayes Decision Theory: Bayes decision rule, Minimum error rate classification, Normal density and discriminant functions
- Parameter Estimation: Maximum Likelihood and Bayesian Parameter Estimation

- **Fractal II: Unsupervised Learning**
- Discriminative Methods: Distance-based methods, Linear Discriminant Functions, Decision Tree, Random Decision Forest and Boosting
- Feature Selection and Dimensionality Reduction: PCA, LDA, ICA, SFFS, SBFS
- Clustering: k-means clustering, Gaussian Mixture Modeling, EM-algorithm

- **Fractal III: Kernels and Neural Networks**
- Kernel Machines: Kernel Tricks, SVMs (primal and dual forms), K-SVR, K-PCA
- Artificial Neural Networks: MLP, Backprop, and RBF-Net
- Foundations of Deep Learning: DNN, CNN, Autoencoders (4 lectures)

Books and Study Materials

- Book:
  - Shalev-Shwartz,S., Ben-David,S., (2014), Understanding Machine Learning: From Theory to Algorithms, Cambridge University Press
  - R. O. Duda, P. E. Hart, D. G. Stork (2000), Pattern Classification, Wiley-Blackwell, 2nd Edition.

- Other useful books:
  - Mitchell Tom (1997). Machine Learning, Tata McGraw-Hill
  - C. M. BISHOP (2006), Pattern Recognition and Machine Learning, Springer-Verlag New York, 1st Edition.

- Other books & Online materials:
  - https://see.stanford.edu/Course/CS229
  - https://www.youtube.com/watch?v=GouhgbE5gPk&list=PLdAoL1zKcqTW-uzoSVBNEecKHsnug_M0k
  - https://www.youtube.com/playlist?list=PLIg1dOXc_acbdJo-AE5RXpIM_rvwrerwR
  - https://www.youtube.com/playlist?list=PL1xHD4vteKYVpaIiy295pg6_SY5qznc77
  - https://www.cs.cmu.edu/~tom/mlbook.html
  - https://people.csail.mit.edu/dsontag/courses/ml16/

Course Logistics

- Instructor: Angshuman Paul
- Contact: apaul@iitj.ac.in
- TA: Jayant Mahawar, Aditi Baheti, Pravin Kumar, Mohit Sharma
- Class hours:
  - Tuesday: 8-8:50 AM
  - Thursday: 8-8:50 AM
  - Friday: 5-5:50 PM
- Office Hours: Will be announced soon

Course Logistics

- Course project (10%)
- Quiz (10%)
- Assignment (10%)
- Viva (10%)
- Minor (15%)
- Major (40%)
- Class Notes and interaction (5%)
- Slides will be shared on every Monday

Course Logistics

- Course project
  - Group of 2 or 3
    - Groups must be formed by 8/8/24
  - You may choose your own project (needs approval)
  - You may choose from a list of projects (FCFS)
  - Total two reports
    - One intermediate report on Sep 10
    - One Final report+ code + other materials on November 10
    - Course Project Viva: November 11-13

Prerequisites

- Basics of
  - Linear Algebra
  - Calculus
  - Probability

What is Machine Learning?

What is Learning

- Learning is any process by which a system improves performance from experience
  - Herbert Simon

- Human improves performance with experience for most problems
  - Example: identification of different animals

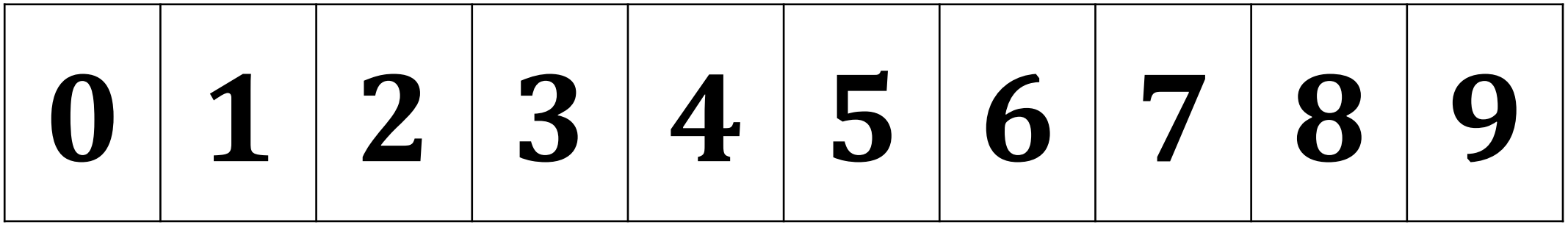
What is Machine Learning

- Definition by Tom Mitchell (1998):
  - Machine Learning is the study of algorithms that
    - improve their performance P
    - at some task T
    - with experience E.
  - A well-defined learning task is given by <P, T, E>

Why Machine Learning?

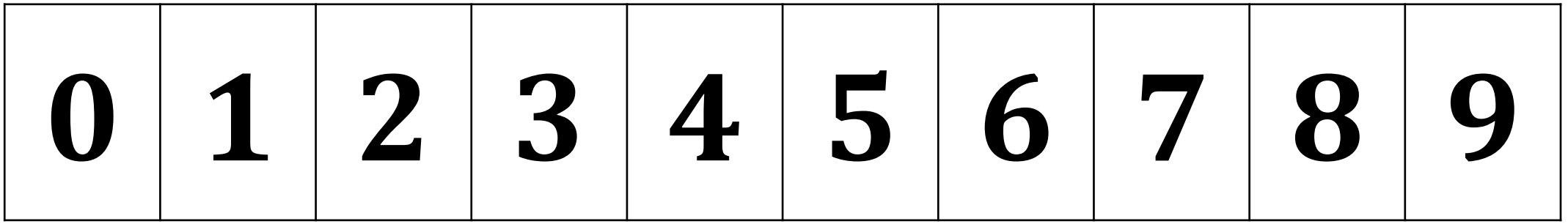
Why ML?

- Suppose, I want to identify different digits from hand-written text



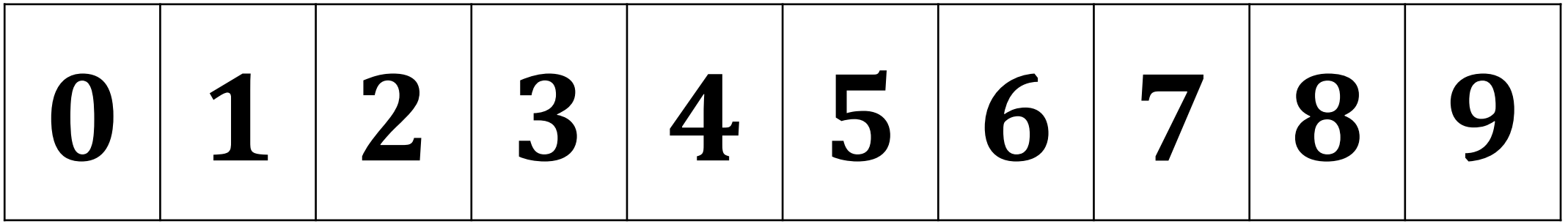
Why ML?

- Suppose, I want to identify different digits from hand-written text
  - How can you program a computer to do this?



Why ML?

- Suppose, I want to identify different digits from hand-written text
  - How can you program a computer to do this?
    - Maybe using shape?



Why ML? An Example

Why ML? An Example

Why ML? An Example

Shape alone may not be sufficient for this problem

Why ML? Another Example

How to differentiate?

https://www.business-standard.com/india-news/happy-international-zebra-day-2024-history-importance-quotes-and-wishes-124013100260_1.html

https://en.wikipedia.org/wiki/Tiger

Why ML? Another Example

How to differentiate?

Body colour?

https://www.business-standard.com/india-news/happy-international-zebra-day-2024-history-importance-quotes-and-wishes-124013100260_1.html

https://en.wikipedia.org/wiki/Tiger

Why ML? Another Example

How to differentiate?

Body colour?

Why ML? Another Example

How to differentiate?

Body colour? Then, what is this animal?

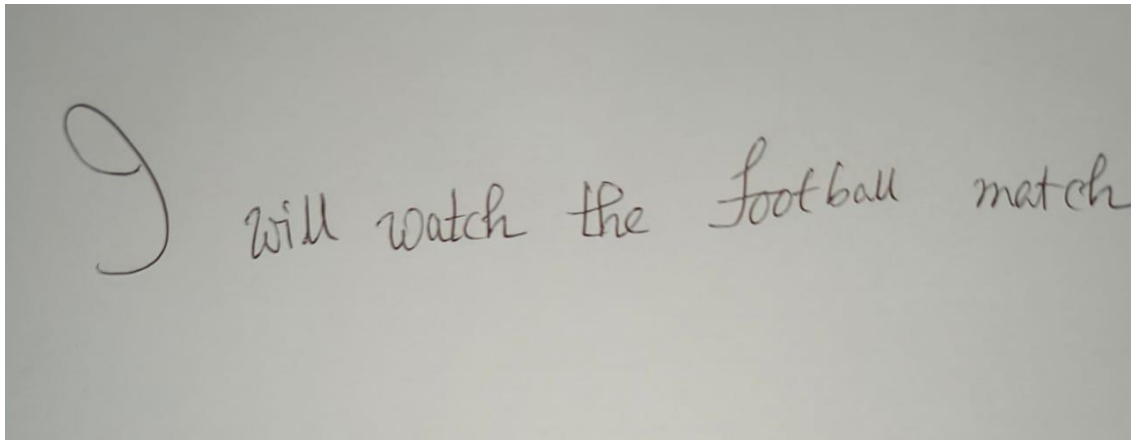
https://en.wikipedia.org/wiki/White_tiger

Why ML? Another Example

Often, we don’t know which feature to use

https://en.wikipedia.org/wiki/White_tiger

Why ML? Another Example

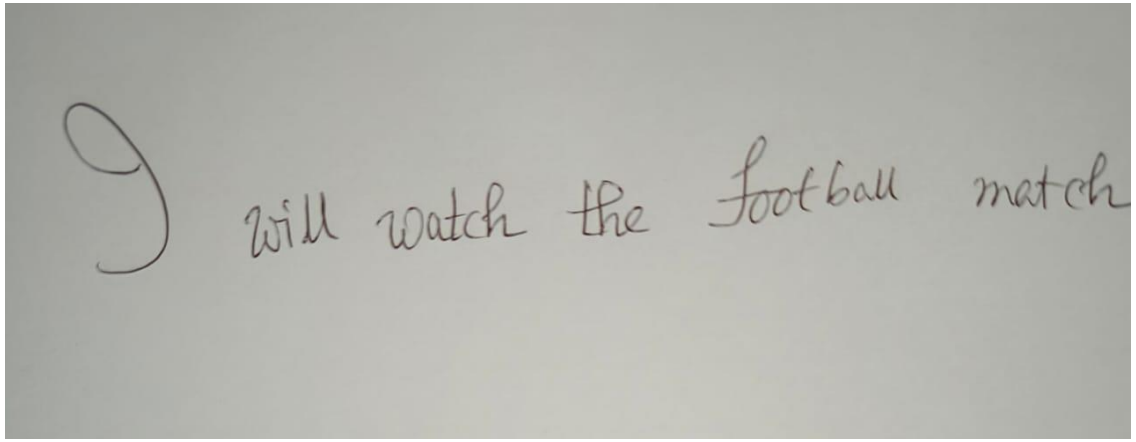


Suppose, I am designing an OCR system

If we try to identify letters from shape (rule-based), it may output

I will watch the fooEball match

Why ML? Another Example



Suppose, I am designing an OCR system

If we try to identify letters from shape (rule-based), it may output

I will watch the footba**ll match**

So, we need to capture the context from data

Why ML?

- We cannot program a computer to do all the tasks of our interest in a rule-based manner (almost always)

- So, probably, a better choice would be to let the computer learn from data

Why ML?

- Automation
  - Automated decision making
- Speed
- Observer independence
  - Through decision support systems

What is Machine Learning

- Definition by Arthur Samuel (1959):
  - Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed.

What is Machine Learning

- Definition by Tom Mitchell (1998):
  - Machine Learning is the study of algorithms that
    - improve their performance P
    - at some task T
    - with experience E.
  - A well-defined learning task is given by <P, T, E>

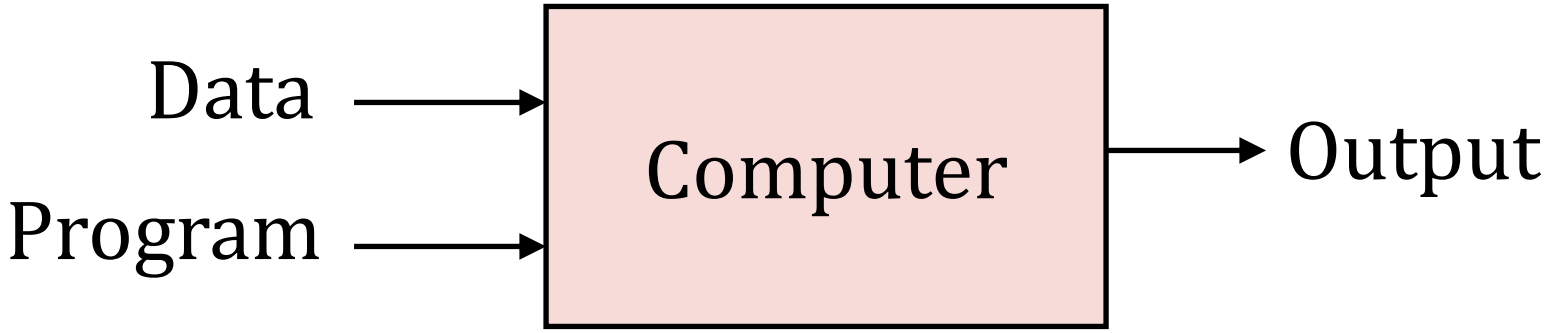
What is Machine Learning

- Definition by Tom Mitchell (1998):
  - Machine Learning is the study of algorithms that
    - improve their performance P
    - at some task T
    - with experience E.
  - A well-defined learning task is given by <P, T, E>

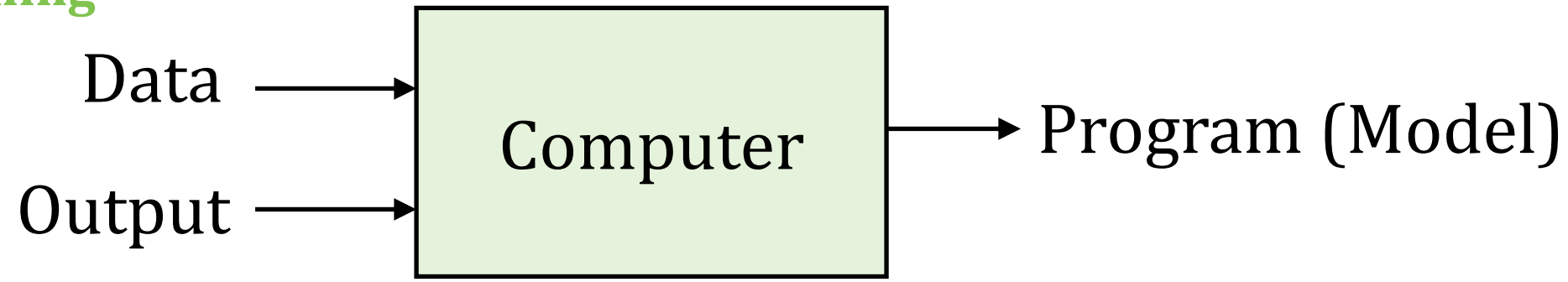
  - **Learning from experience: Inductive learning**

What is Machine Learning?

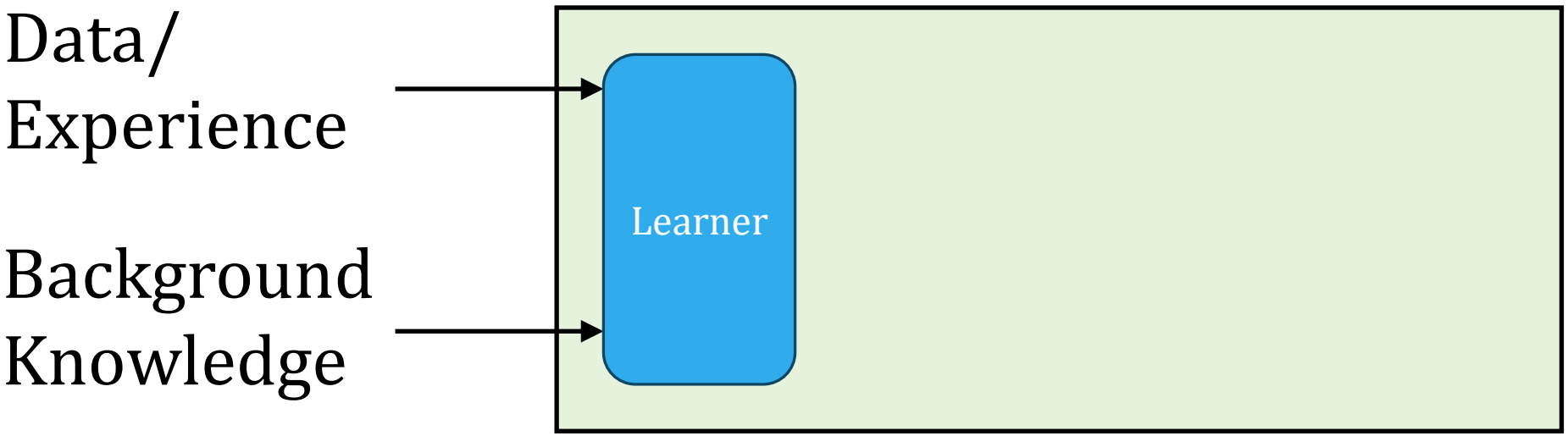
Traditional Programming



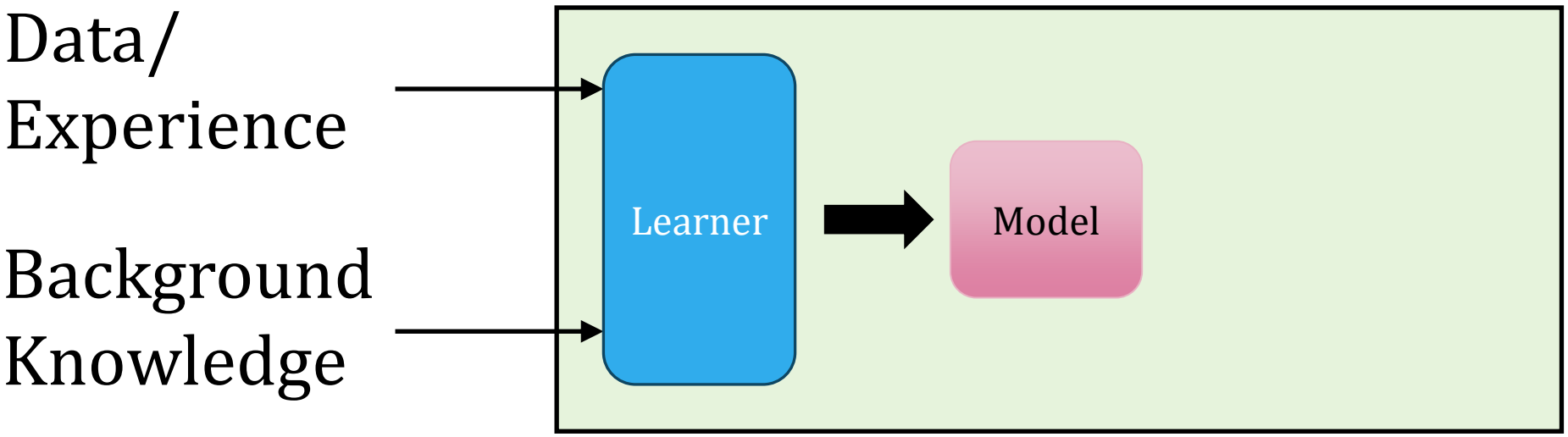
Machine Learning



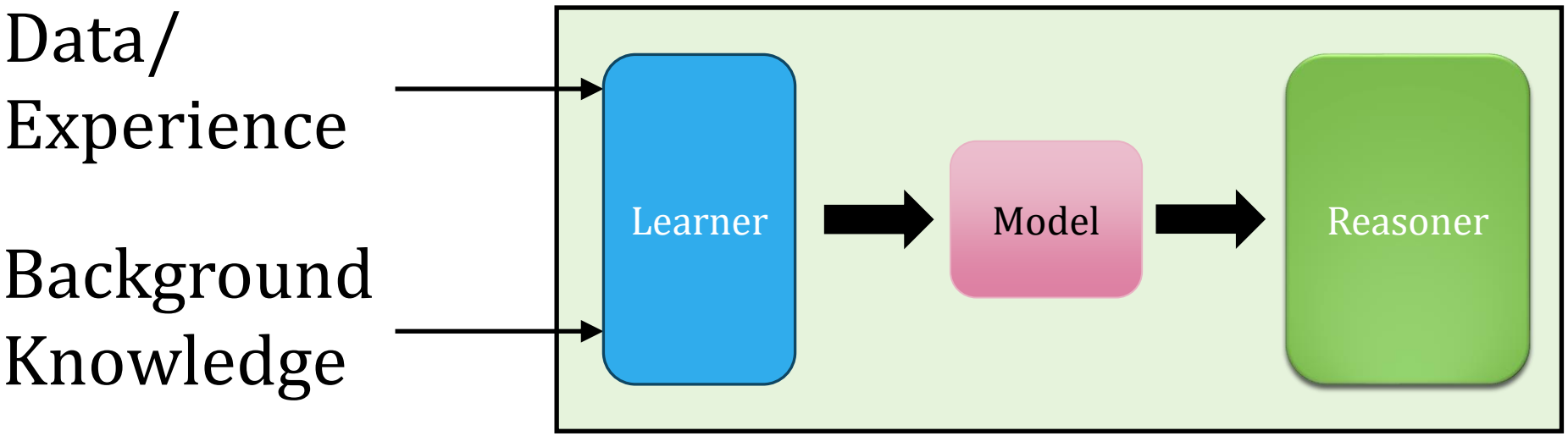
What is Machine Learning?



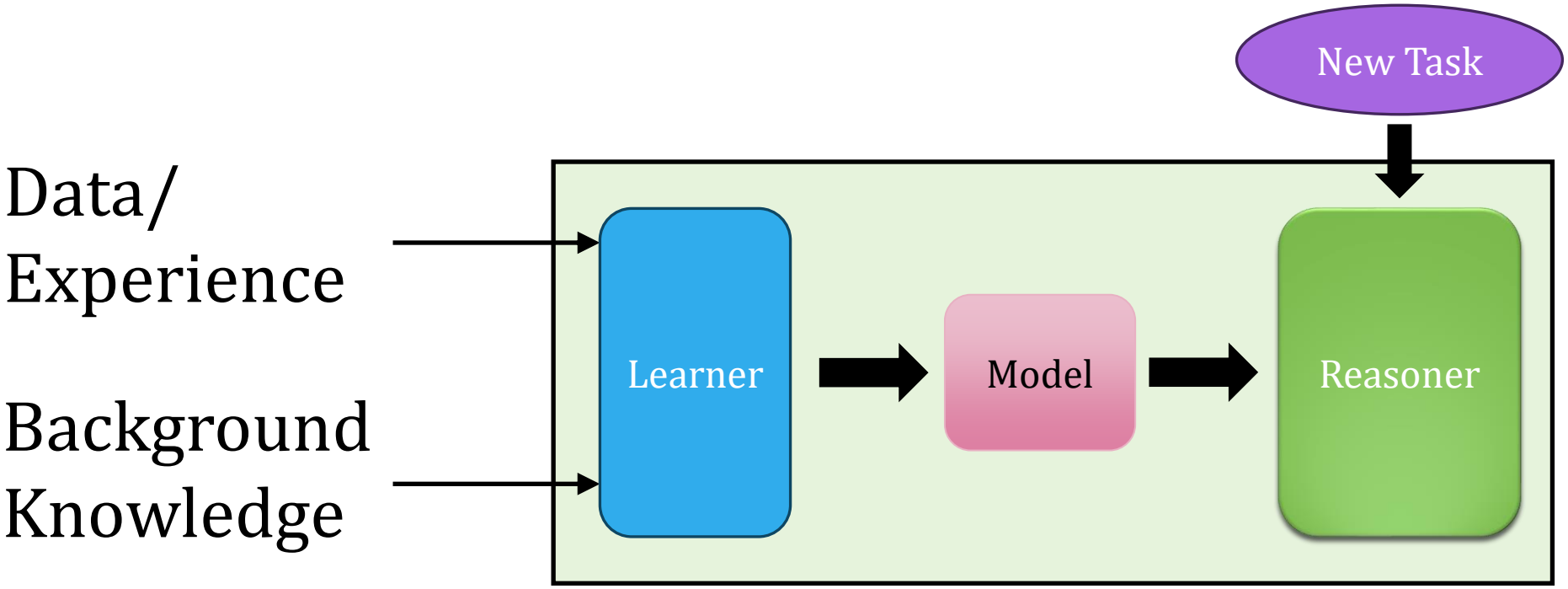
What is Machine Learning?



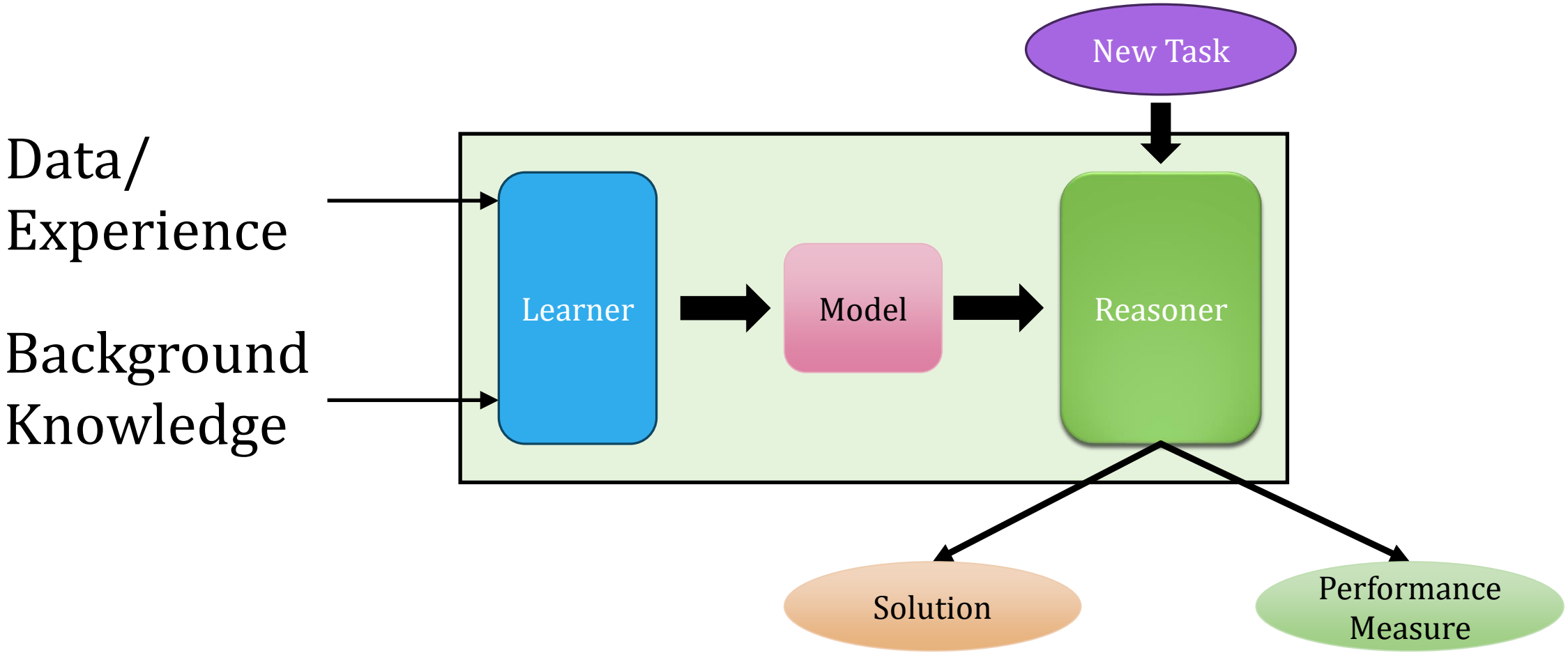
What is Machine Learning?



What is Machine Learning?



What is Machine Learning?



Steps for Creating a Learner

- Choose the training experience/ training data
  - Extract features from data (feature extraction may not be required for all methods)
- Choose the target function
  - The function to be learnt (e.g., in cat v dog image classification, we want a function that results in different set of values for cat and dog)
- Choose how to represent the target function
  - e.g., do we want the target function to be linear or circular or something else?
- Choose a learning algorithm to infer the target function

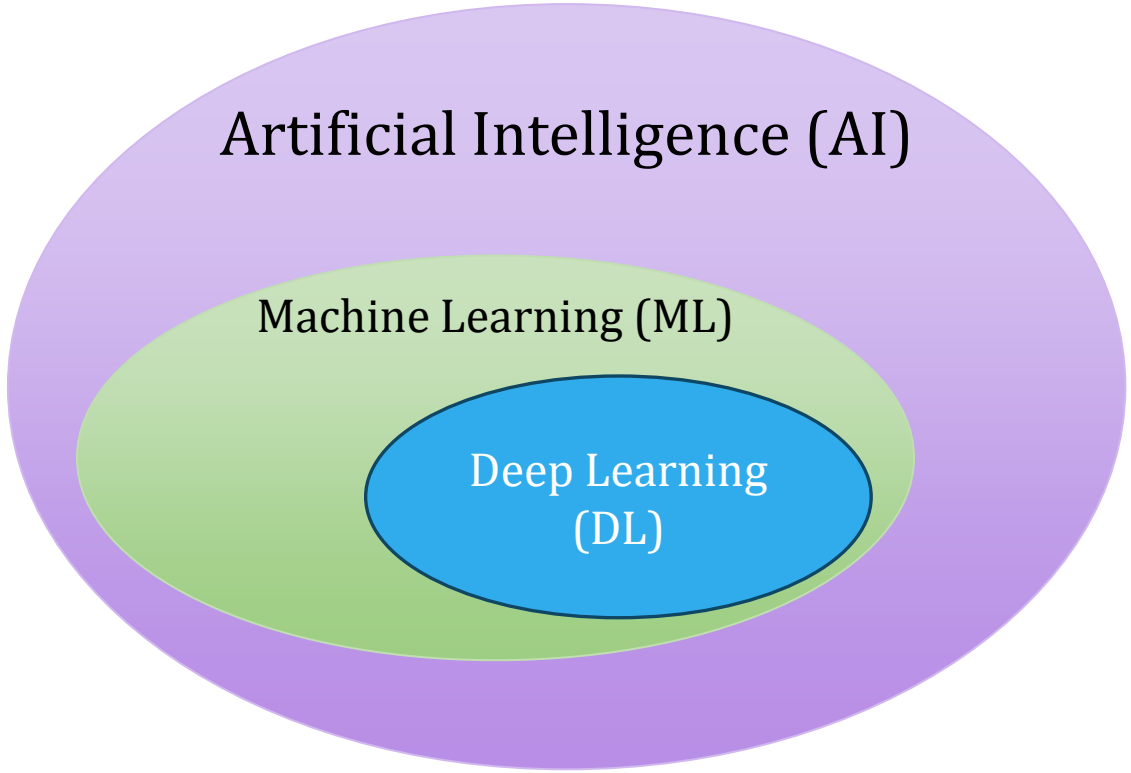
Steps for Creating a Learner

- Choose the training experience/ training data
  - Extract features from data (feature extraction may not be required for all methods)
- Choose the target function
  - The function to be learnt (e.g., in cat v dog image classification, we want a function that results in different set of values for cat and dog)
- Choose how to represent the target function
  - A rich representation
    - May represent complex function
    - May be difficult to learn
- Choose a learning algorithm to infer the target function

Steps for Creating a Learner

- Choose the training experience/ training data
  - Extract features from data (feature extraction may not be required for all methods)
- Choose the target function
  - The function to be learnt (e.g., in cat v dog image classification, we want a function that results in different set of values for cat and dog)
- Choose how to represent the target function
  - Class of functions: Hypothesis language
- Choose a learning algorithm to infer the target function

What is Machine Learning



History of ML

- Used to be known as pattern recognition (PR)

- The term ***Machine Learning*** was coined in1959 by Arthur Samuel, an IBM employee and pioneer in the field of computer gaming and artificial intelligence

History of ML

- 1950s:
  - Samuel's checker-playing program
- 1960s:
  - Neural network: Rosenblatt's perceptron
  - Pattern Recognition
  - Minsky & Papert prove limitations of Perceptron
- 1970s:
  - Symbolic concept induction
  - Expert systems and knowledge acquisition bottleneck
  - Quinlan's ID3
  - Natural language processing (symbolic)

https://www.youtube.com/watch?v=T3PsRW6wZSY&list=PLIg1dOXc_acbdJo-AE5RXpIM_rvwrerwR

History of ML

- 1980s:
  - Advanced decision tree and rule learning
  - Learning and planning and problem solving
  - Resurgence of neural network
  - Valiant's PAC learning theory
  - Focus on experimental methodology
- 90's ML and Statistics
  - Support Vector Machines
  - Data Mining
  - Adaptive agents and web applications
  - Text learning
  - Reinforcement learning
  - Ensembles
  - Bayes Net learning

- 1994: Self-driving car road test
- 1997: Deep Blue beats Gary Kasparov

https://www.youtube.com/watch?v=T3PsRW6wZSY&list=PLIg1dOXc_acbdJo-AE5RXpIM_rvwrerwR

History of ML

- 1980s:
  - Advanced decision tree and rule learning
  - Learning and planning and problem solving
  - Resurgence of neural network
  - Valiant's PAC learning theory
  - Focus on experimental methodology
- 90's ML and Statistics
  - Support Vector Machines
  - Data Mining
  - Adaptive agents and web applications
  - Text learning
  - Reinforcement learning
  - Ensembles
  - Bayes Net learning

- 1994: Self-driving car road test
- 1997: Deep Blue beats Gary Kasparov

https://www.youtube.com/watch?v=T3PsRW6wZSY&list=PLIg1dOXc_acbdJo-AE5RXpIM_rvwrerwR

History of ML

- Popularity of this field in recent time and the reasons behind that
  - New software/ algorithms
    - Neural networks
    - Deep learning
  - New hardware
    - GPU's
  - Cloud Enabled
  - Availability of Big Data

- 2009: Google builds self driving car
- 2011: Watson wins Jeopardy
- 2014: Human vision surpassed by ML systems

https://www.youtube.com/watch?v=T3PsRW6wZSY&list=PLIg1dOXc_acbdJo-AE5RXpIM_rvwrerwR

History of ML

https://medium.com/@marizombie/important-dates-in-history-of-machine-learning-d52d7f24b797

ML Today

Massachusetts Institute of Technology

Education Research Innovation Admissions + Aid Campus Life News Alumni About MIT

MIT News

ON CAMPUS AND AROUND THE WORLD

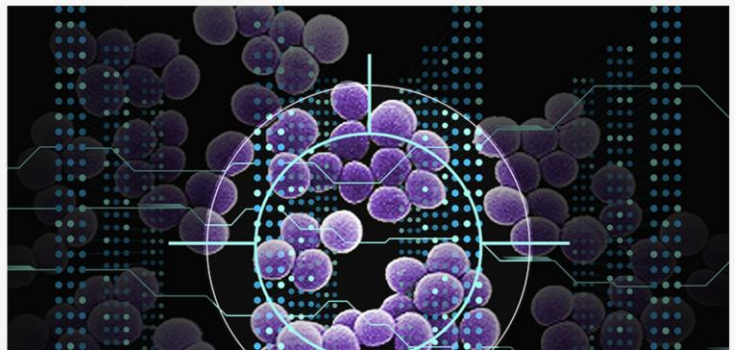
SUBSCRIBE | BROWSE | SEARCH NEWS

Using AI, MIT researchers identify a new class of antibiotic candidates

These compounds can kill methicillin-resistant Staphylococcus aureus (MRSA), a bacterium that causes deadly infections.

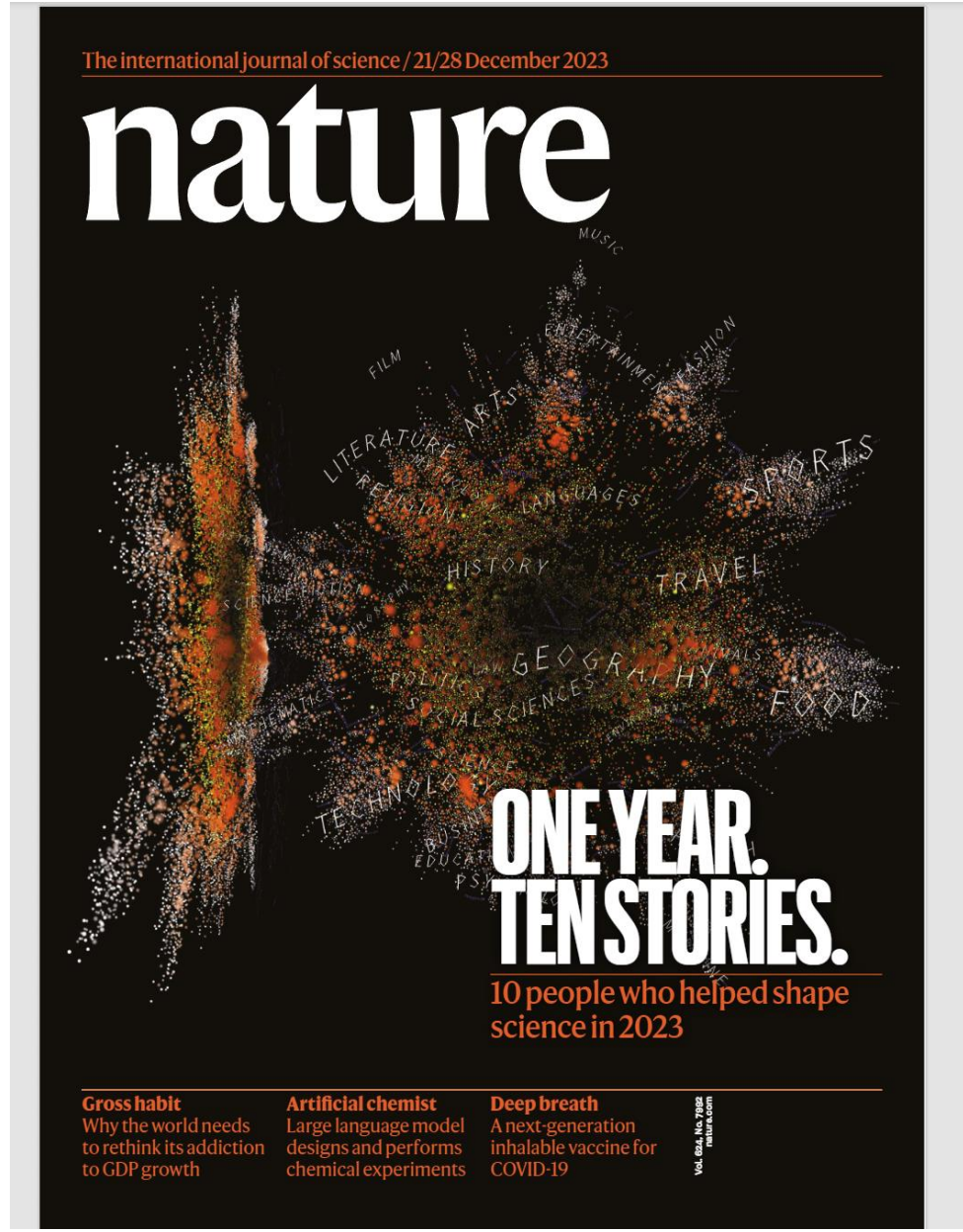
Anne Trafton | MIT News
December 20, 2023

PRESS INQUIRIES



Using a type of artificial intelligence known as deep learning, MIT researchers have discovered a class of compounds that can kill a drug-resistant bacterium that causes more than 10,000 deaths in the United States every year.

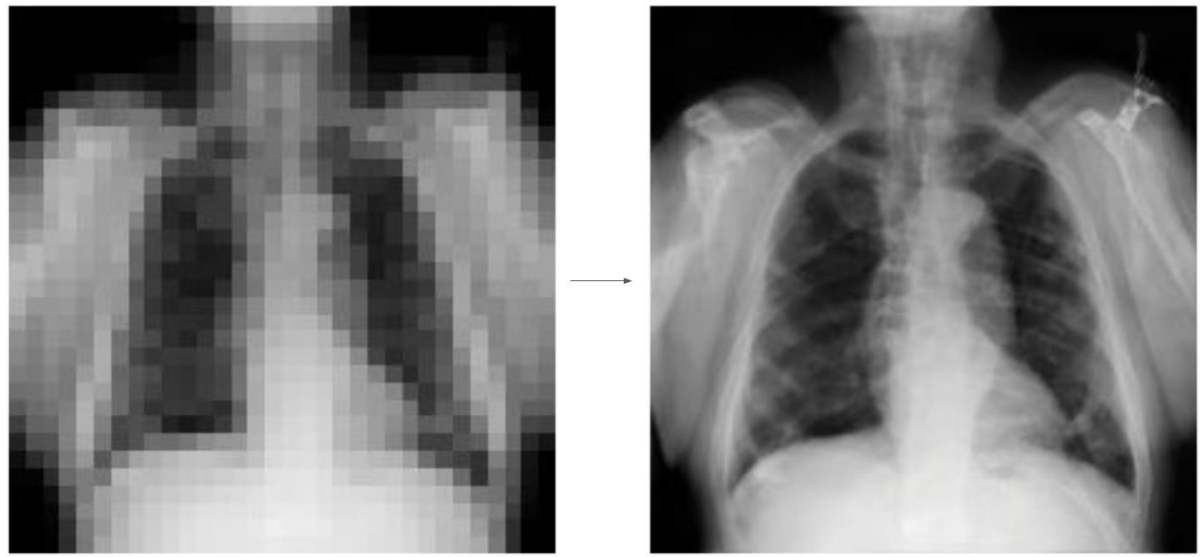
Image: Christine Daniloff, MIT; Janice Haney Carr, CDC; iStock



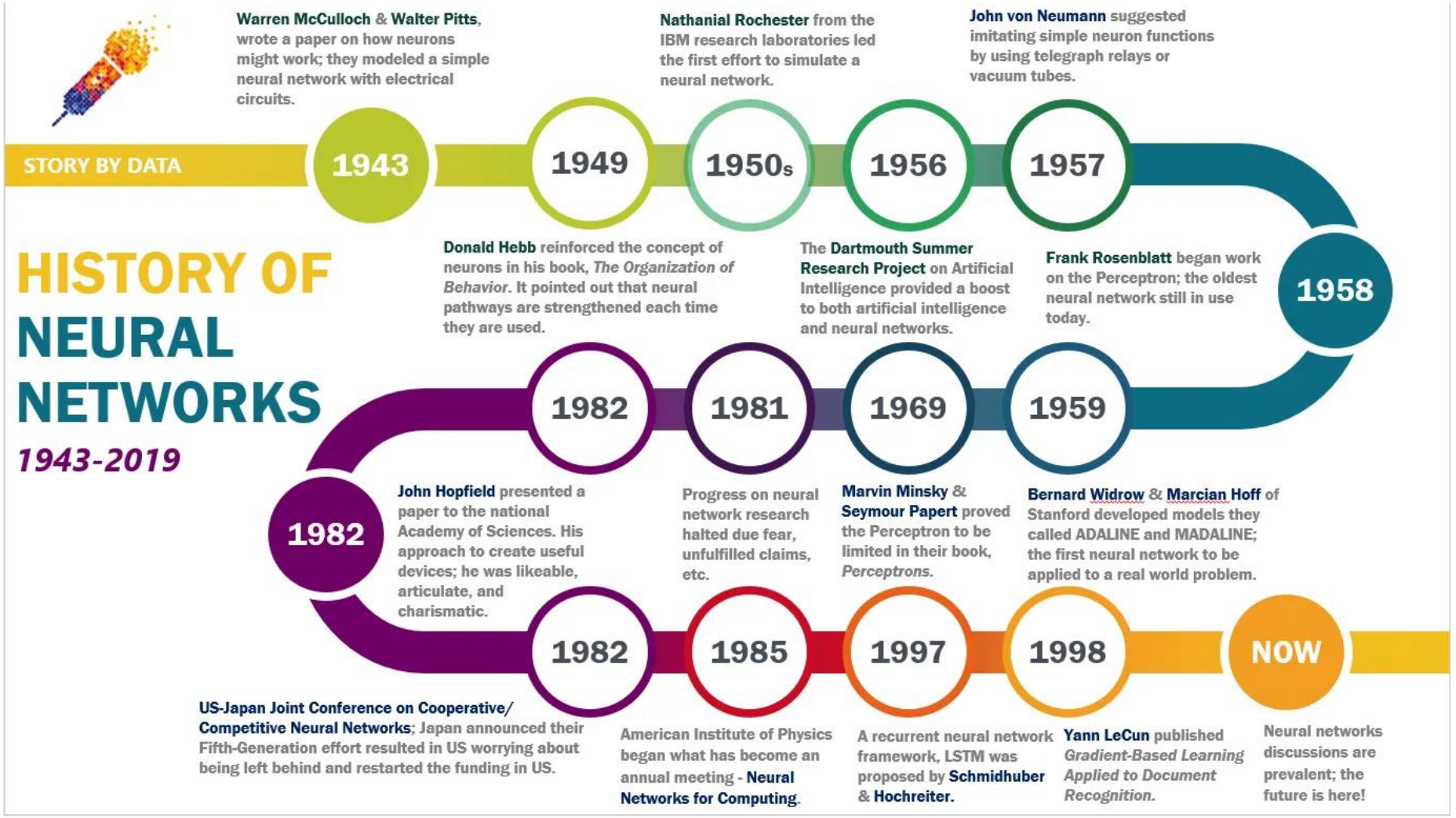
2. AI finally starting to feel like AI

The stratospheric ascent of OpenAI's ChatGPT this year has prompted furious debate in the media and elsewhere about the future role of artificial intelligence and its implications for everything from employment to healthcare. Photograph: Lionel Bonaventure/AFP/Getty Images

It's often hard to spot technological watersheds until long after the fact, but 2023 is one of those rare years in which we can say with certainty that the world changed. It was the year in which artificial intelligence (AI) finally went mainstream. I'm referring, of course, to ChatGPT and its stablemates - large language models. Released late in 2022, ChatGPT went viral in 2023, dazzling users with its fluency and seemingly encyclopaedic knowledge. The tech industry - led by trillion-dollar companies - was wrong-footed by the success of a product from a company with just a few hundred employees. As I write, there is desperate jostling to take the lead in the new "generative AI" marketplace heralded by ChatGPT.



A Brief History of Neural Networks



https://medium.com/analytics-vidhya/brief-history-of-neural-networks-44c2bf72eec#:~:text=The%20first%20step%20towards%20neural,neural%20network%20with%20electrical%20circuits.

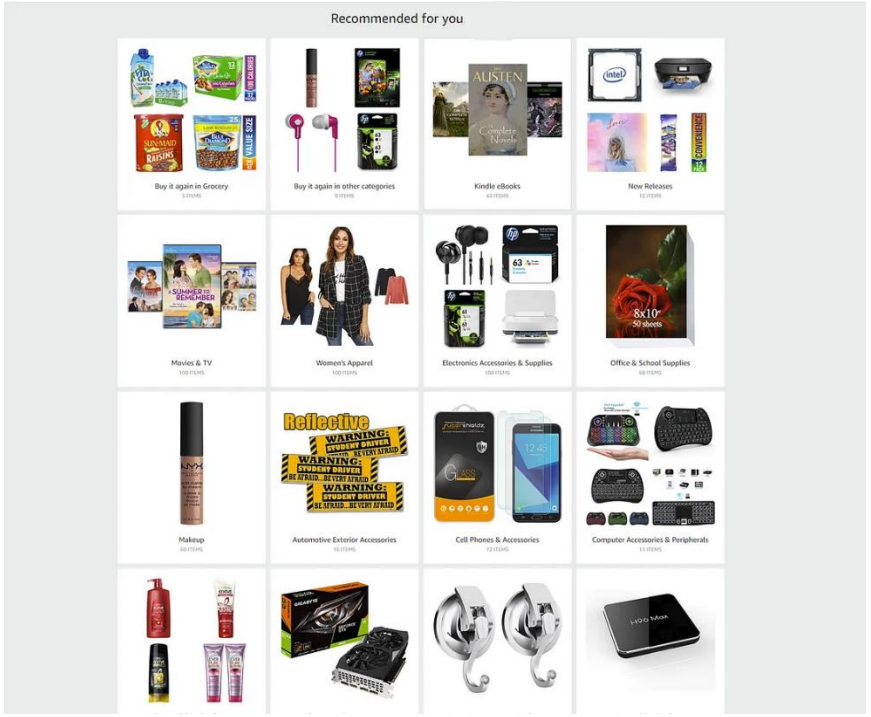
Is it Only the Success Story?

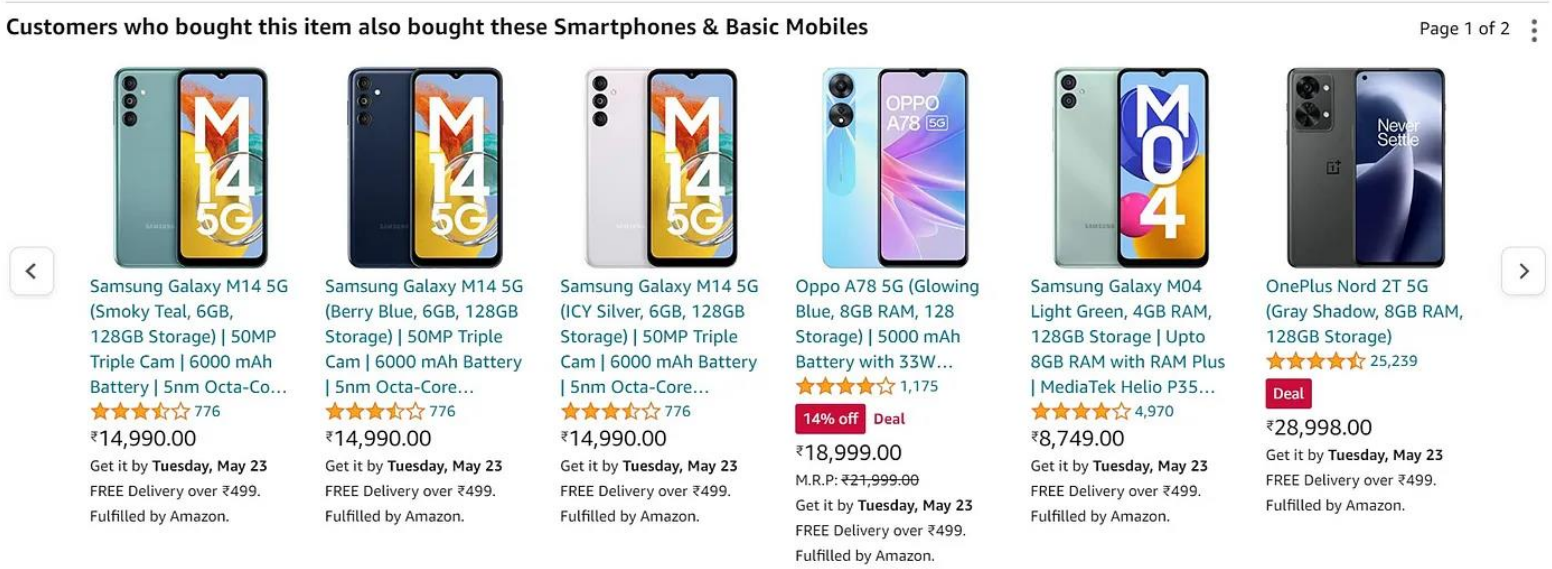
- Explainability
- Generalizability
- Computational requirements
- Data requirements

What's New Today?

- Availability of hardware
  - GPUs
  - Storage
  - Cloud services
- Availability of large datasets
- Availability of implementation platforms
  - PyTorch
  - Tensorflow

Applications of ML

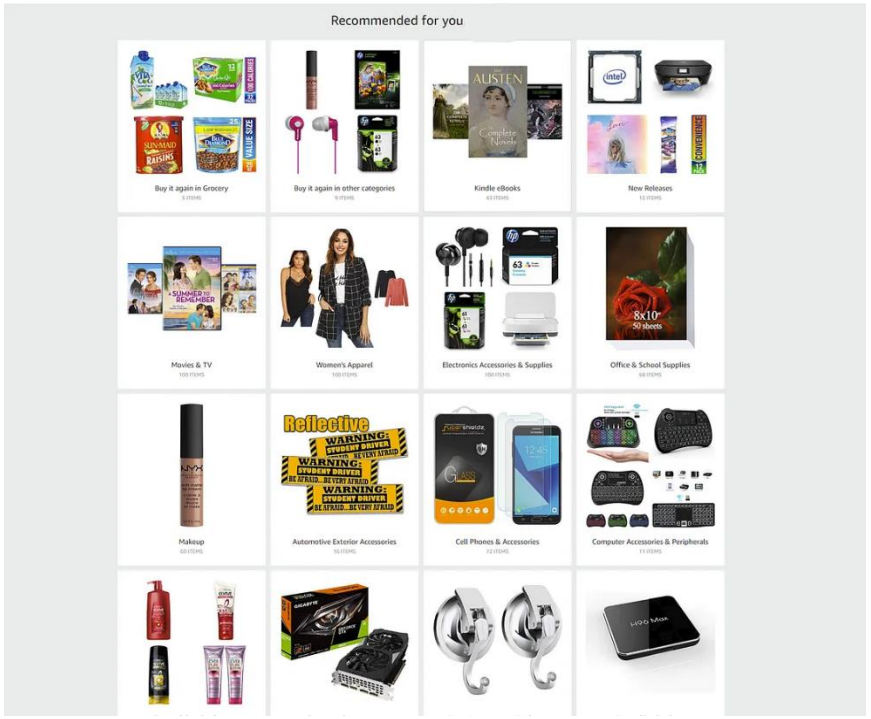


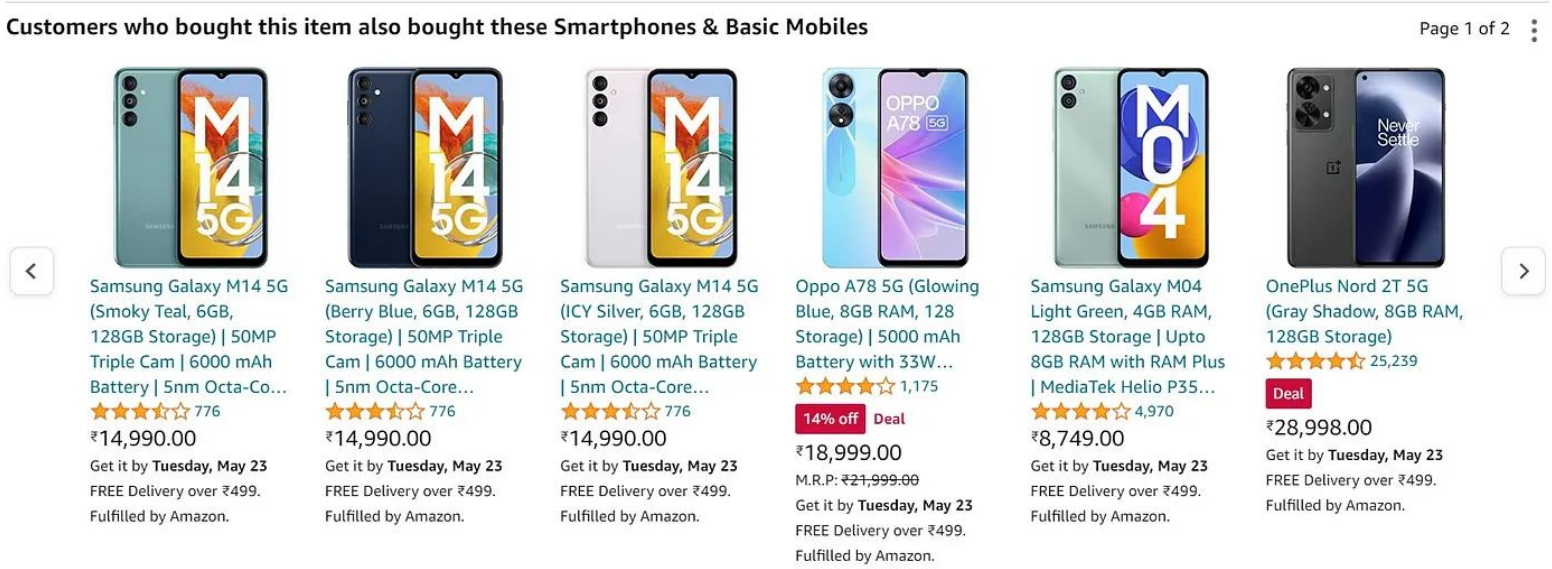


https://towardsdatascience.com/product-recommender-using-amazon-review-dataset-e69d479d81dd

https://rhydhamgupta.medium.com/building-a-recommender-system-from-scratch-1-dd140492afa6

Applications of ML





Recommendation System

https://towardsdatascience.com/product-recommender-using-amazon-review-dataset-e69d479d81dd

https://rhydhamgupta.medium.com/building-a-recommender-system-from-scratch-1-dd140492afa6

Applications of ML

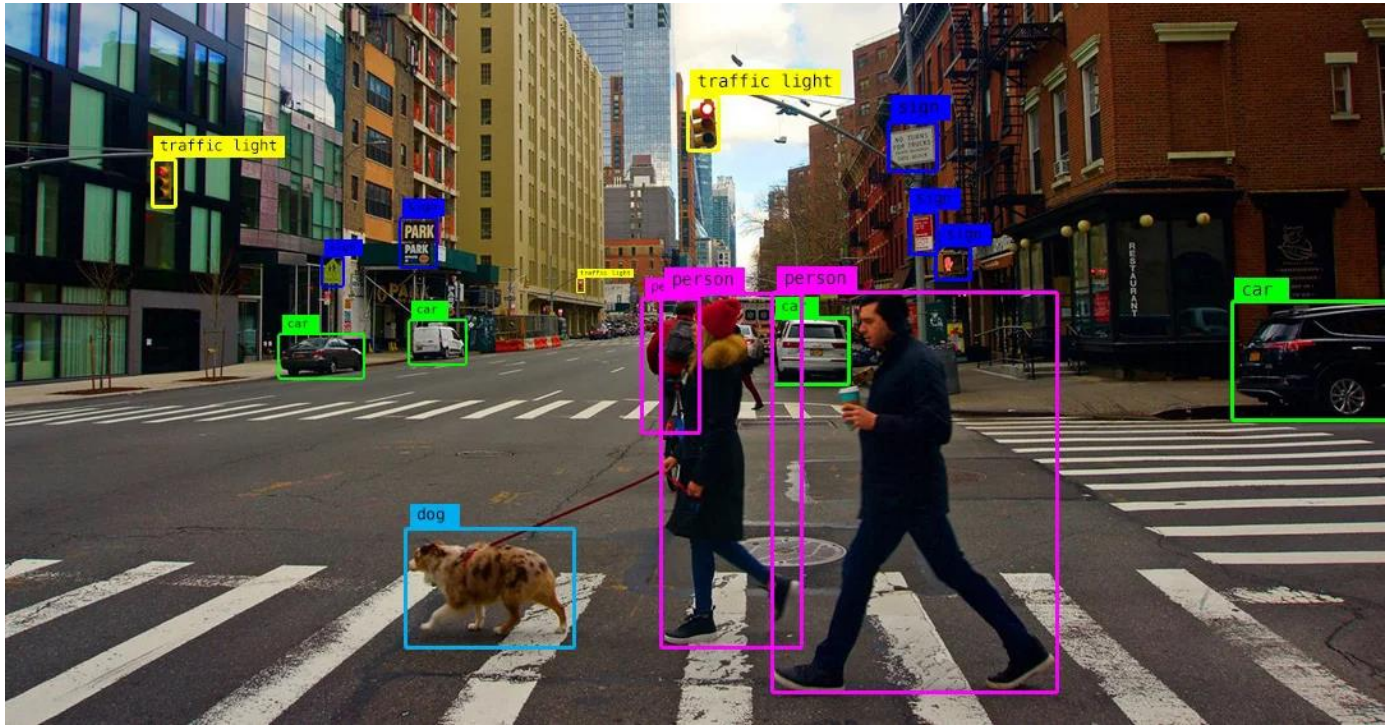
Autocomplete

Applications of ML

Speech recognition

https://www.appen.com/blog/what-is-automatic-speech-recognition

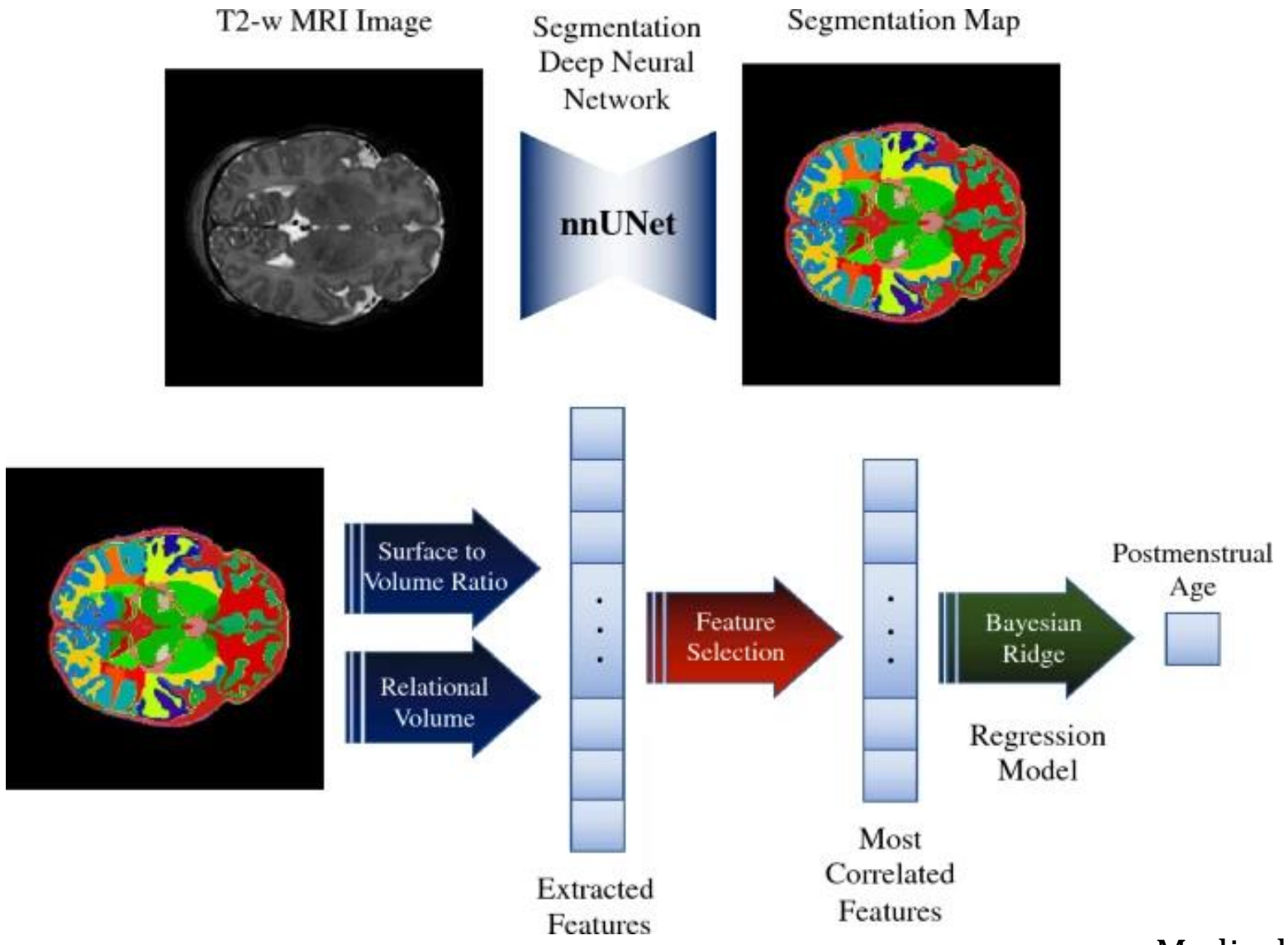
Applications of ML



Computer Vision

https://www.algotive.ai/blog/what-is-computer-vision-and-how-does-it-work-with-artificial-intelligence

Applications of ML



https://www.nature.com/articles/s41598-023-40244-z

Medical Image Analysis

Applications of ML

- Medicine
- Robotics
- Biometrics
- Finance & banking
- Physics
- Security
- And many more

Different Types of ML

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

Different Types of ML

- **Supervised Learning**
- Unsupervised Learning
- Reinforcement Learning

Supervised Learning

- Experience

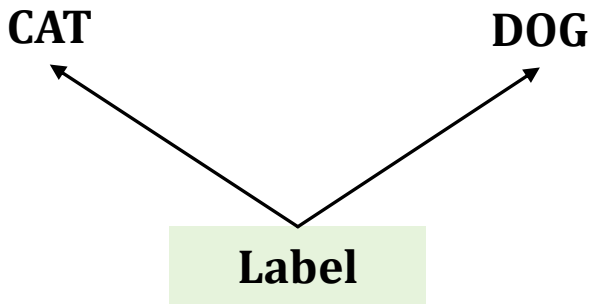
CAT

DOG

https://www.pexels.com/

Supervised Learning

- Experience

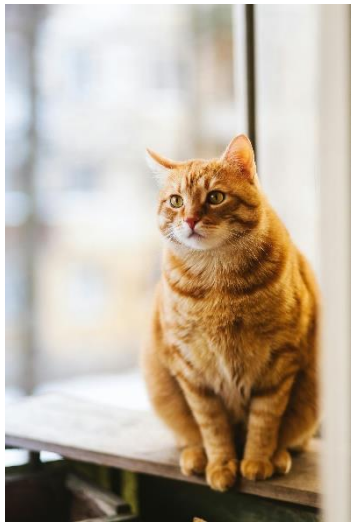


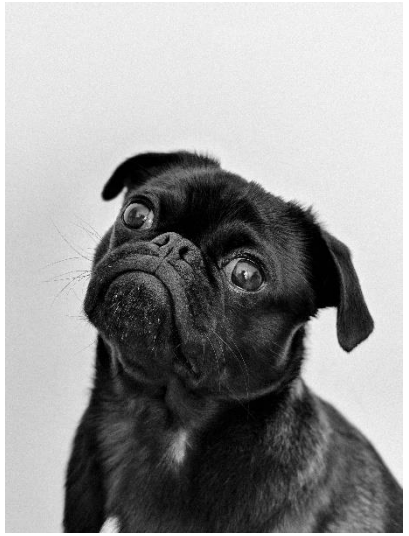
Data

Data

Supervised Learning

- Task: Predict the type of the animals





Cat or Dog?

https://www.pexels.com/

Supervised Learning

- Experience

Data

https://www.pexels.com/

Data

Supervised Learning

- Experience

Data

Price: 3.4 Cr

Price: 1.8 Cr

Price: 1.2 Cr

Label

Supervised Learning

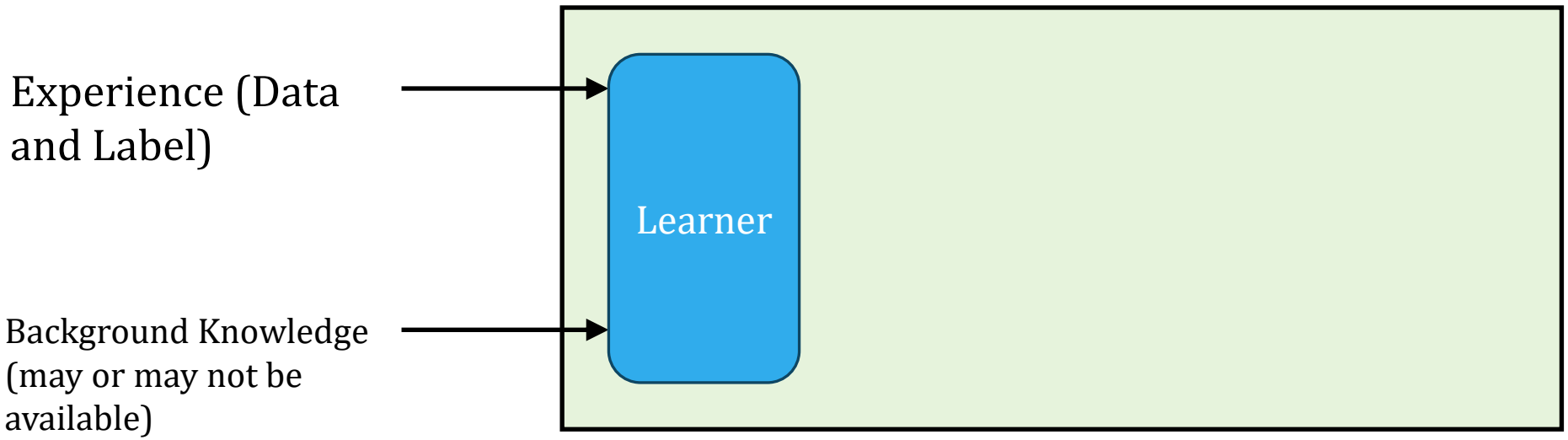
- Task: Predict the price of this house

https://www.pexels.com/

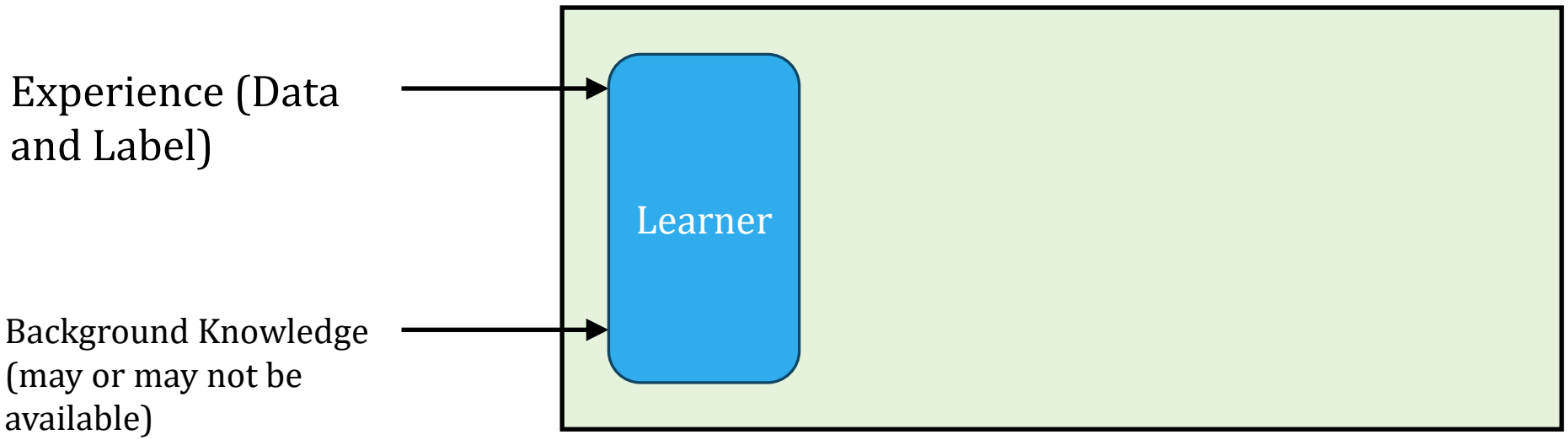
Supervised Learning

- Input to learner (input for experience)
  - Data and label
- Creation of model using experience: training
- Once the model is ready
  - Reasoner comes into picture
  - Input to the reasoner is only the data
  - The reasoner predicts the label

Supervised Learning

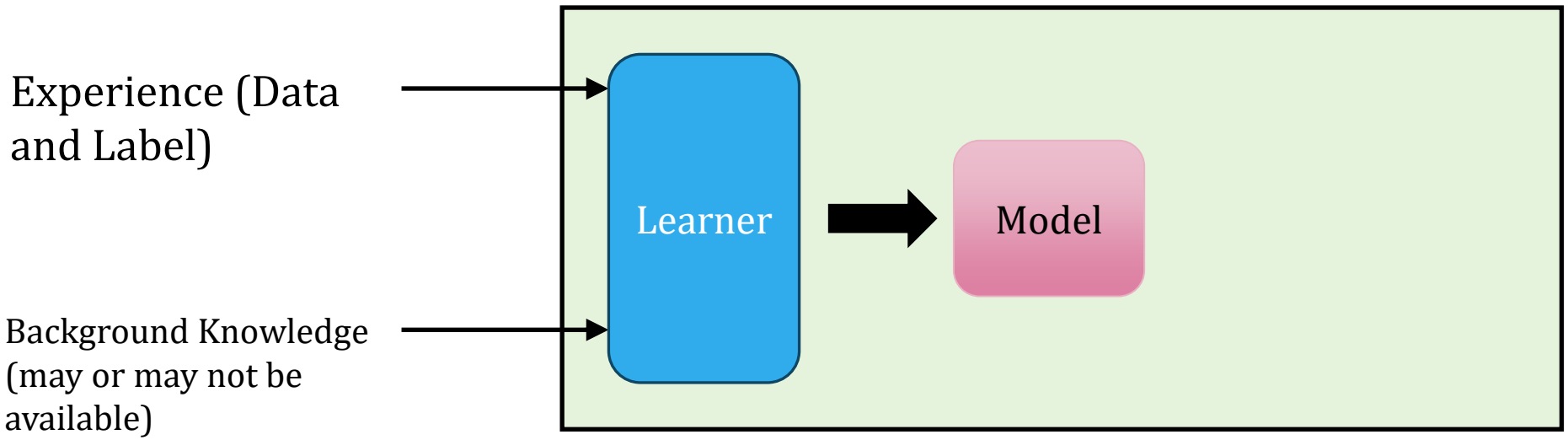


Supervised Learning



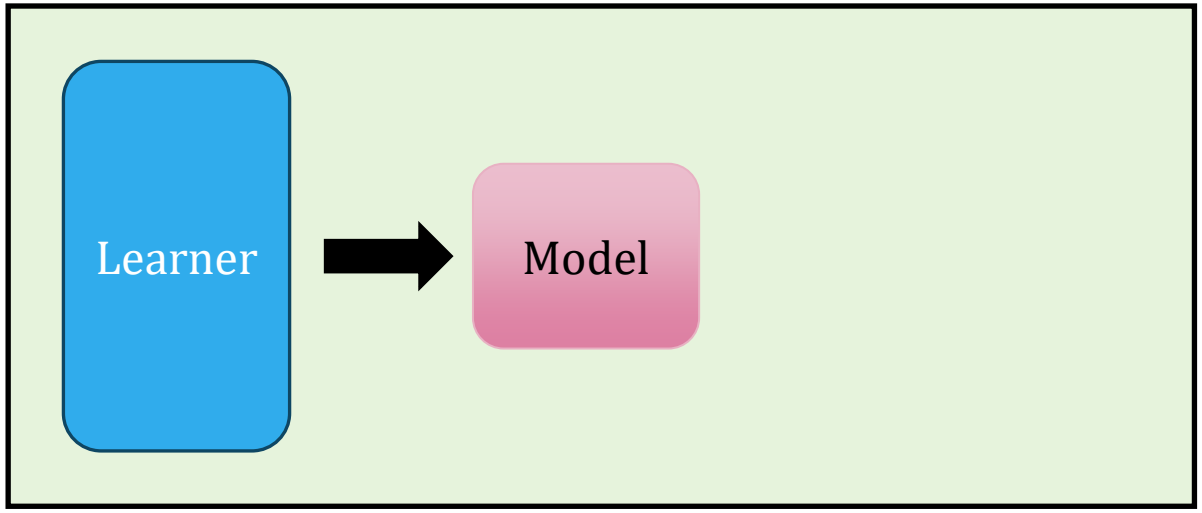
Training

Supervised Learning



End of Training

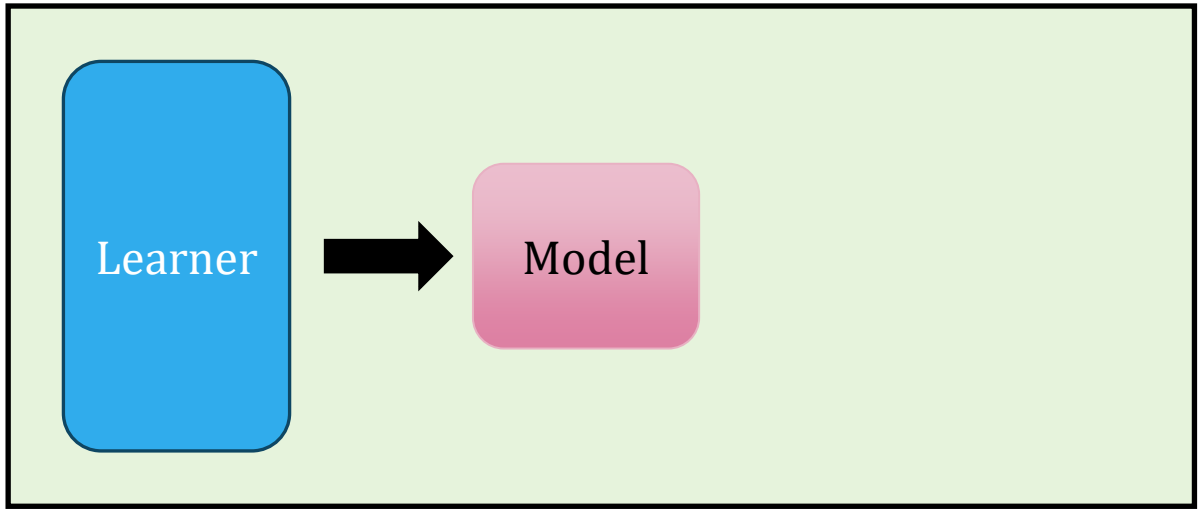
Supervised Learning



Validation for tuning the hyperparameters of the model

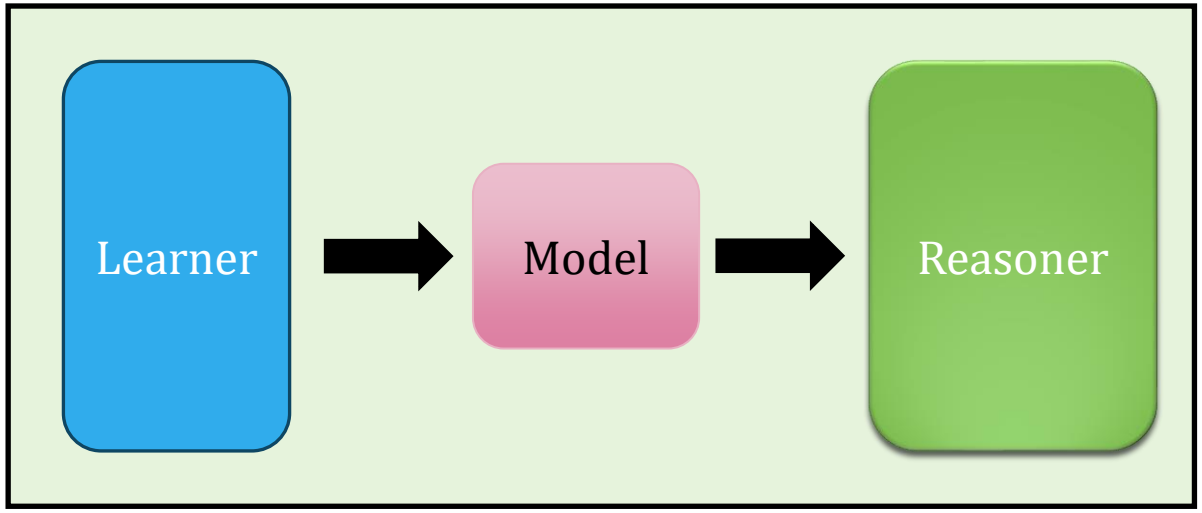
Must be performed using validation data (and not test data)

Supervised Learning

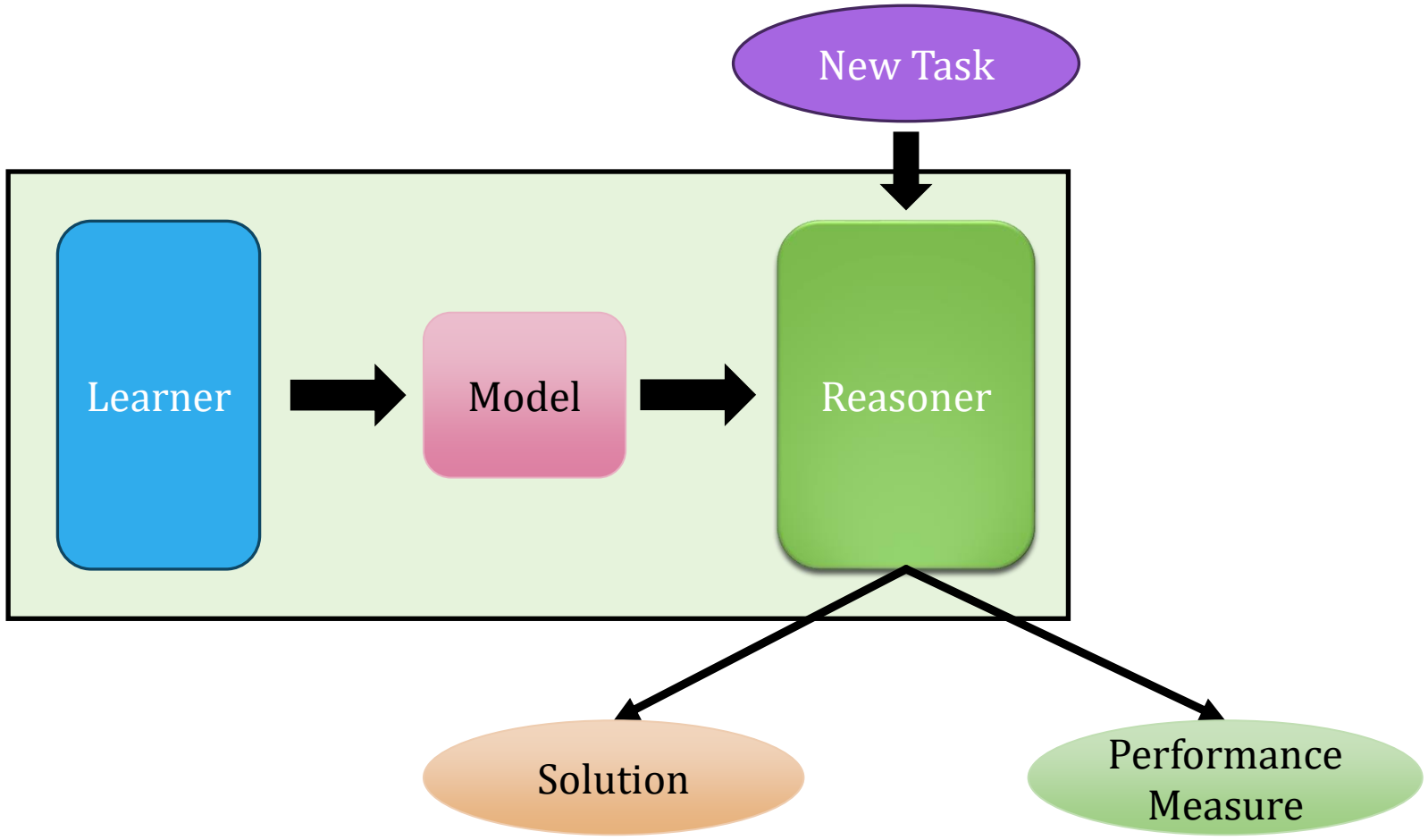


Model ready for deployment

What is Machine Learning?



Supervised Learning

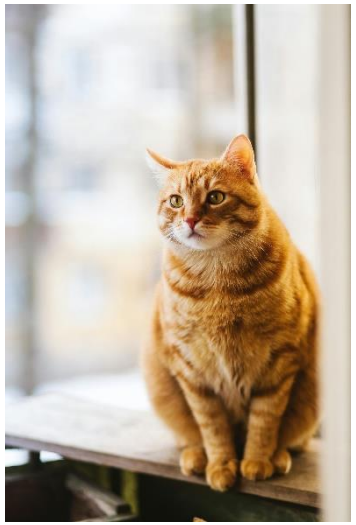


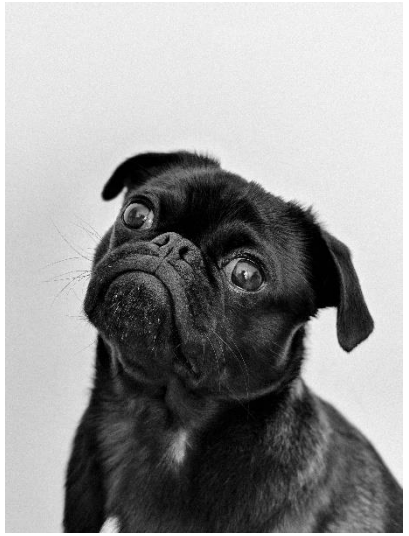
Supervised Learning

- Mainly of two types
  - Classification
    - Labels are categorical in nature
      - (Cat, dog, elephant), (foreground, background), etc.

Supervised Learning

- Task: Predict the type of the animals





Cat or Dog?

https://www.pexels.com/

Supervised Learning

- Mainly of two types
  - Classification
    - Labels are categorical in nature
      - (Cat, dog, elephant), (foreground, background), etc.
  - Regression
    - Labels are real numbers
      - Price of house, Hemoglobin (in gm%), coordinate of a robot, etc.

Supervised Learning

- Task: Predict the price of this house

https://www.pexels.com/

Other Types of Data

(a)

(b)

(c)

Figure 1.1: Three types of Iris flowers: Setosa, Versicolor and Virginica. Used with kind permission of Dennis Kramb and SIGNA.

index	sl	sw	pl	pw	label
0	5.1	3.5	1.4	0.2	Setosa
1	4.9	3.0	1.4	0.2	Setosa
	...				
50	7.0	3.2	4.7	1.4	Versicolor
	...				
149	5.9	3.0	5.1	1.8	Virginica

Table 1.1: A subset of the Iris design matrix. The features are: sepal length, sepal width, petal length, petal width. There are 50 examples of each class.

- Data may be available in its raw form (e.g., image, audio) or may be in the form of a vector representing different features

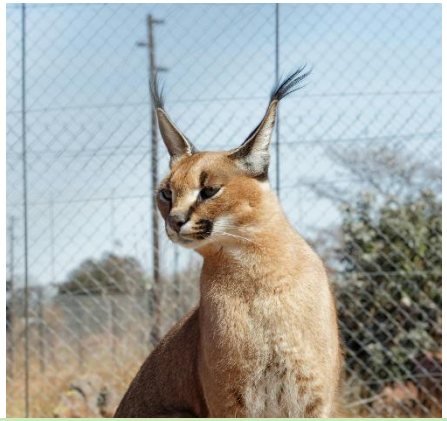
Different Types of ML

- Supervised Learning
- **Unsupervised Learning**
- Reinforcement Learning

Unsupervised Learning

https://www.pexels.com/

Unsupervised Learning



Do you know the names of these animals?

https://www.pexels.com/

Unsupervised Learning

Can you group these animals?

https://www.pexels.com/

Unsupervised Learning

1

2

3

4

5

6

7

https://www.pexels.com/

Unsupervised Learning

Unsupervised Learning

Only data and no labels

Unsupervised Learning

- Making sense of data
- Input to learner (input for experience)
  - Data
  - No label
- Output: Target task such as clustering

Unsupervised Learning Example

- Clustering
  - Making cohesive group of data points
  - Example application
    - Recommendation system based on customer's pervious choices
    - Finding similar viruses based on genetic pattern
- Dimensionality reduction
  - Application: finding salient features in data
  - Data compression
- Association rule mining
  - Finding cooccurrence
- Generative Models
- Unsupervised feature extraction

Clustering

Unsupervised Learning Example

- Clustering
  - Making cohesive group of data points
  - Example application
    - Recommendation system based on customer's pervious choices
    - Finding similar viruses based on genetic pattern
- Dimensionality reduction
  - Application: finding salient features in data
  - Data compression
- Association rule mining
  - Finding cooccurrence
- Generative Models
- Unsupervised feature extraction

Different Types of ML

- Supervised Learning
- Unsupervised Learning
- **Reinforcement Learning**

Reinforcement Learning

How did you learn to ride bicycle?

https://ptscoaching.com/2015/04/teaching-your-child-life-skills/

Reinforcement Learning

- How did you learn to ride bicycle?
- Supervised?
- Unsupervised?

https://ptscoaching.com/2015/04/teaching-your-child-life-skills/

Reinforcement Learning

- How did you learn to ride bicycle?
- Supervised?
- Unsupervised?

Some training is given initially. Then, we gradually learn how much pressure to put in the paddles, how much bending is required to turn the bicycle, where to apply the break to stop at a point

https://ptscoaching.com/2015/04/teaching-your-child-life-skills/

Reinforcement Learning

Some training is given initially. Then, we gradually learn how much pressure to put in the paddles, how much bending is required to turn the bicycle, where to apply the break to stop at a point.

How do we learn these?

From rewards and penalties

Penalty: falling down, getting disbalanced, etc.
Reward: Reaching in time, praise from elders, etc.

https://ptscoaching.com/2015/04/teaching-your-child-life-skills/

Reinforcement Learning

Some training is given initially. Then, we gradually learn how much pressure to put in the paddles, how much bending is required to turn the bicycle, where to apply the break to stop at a point.

How do we learn these?

From rewards and penalties

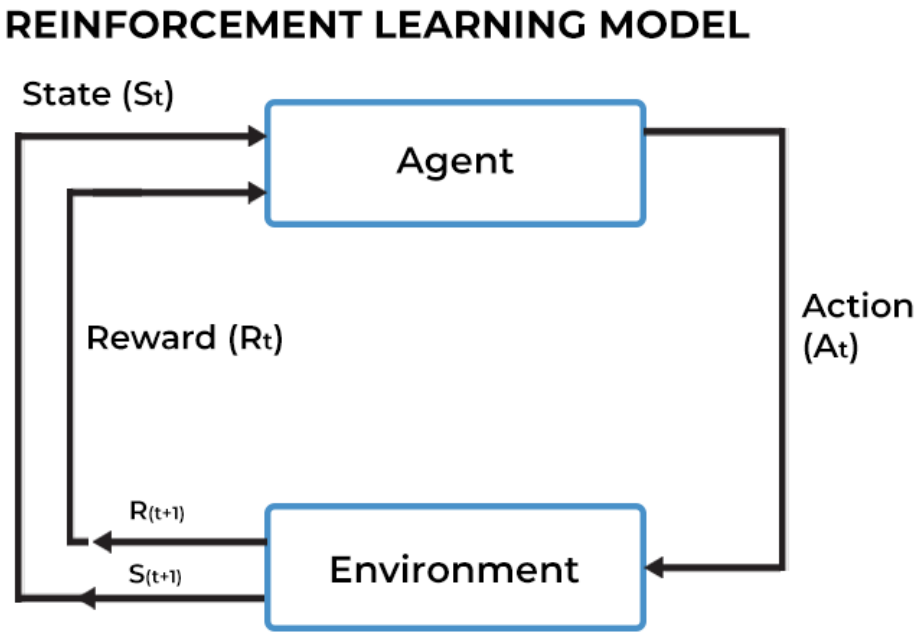
Penalty: falling down, getting disbalanced, etc.
Reward: Reaching in time, praise from elders, etc.

What do we learn?
To control an agent (bicycle) through **a sequence of** good decisions (actions)

How do we learn it?
From experiences through rewards and penalty from the environment

https://ptscoaching.com/2015/04/teaching-your-child-life-skills/

Reinforcement Learning



What do we learn?
To control an agent (bicycle) through **a sequence of** good decisions (actions)

How do we learn it?
From experiences through rewards and penalty from the environment

https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-reinforcement-learning/

Other Types of ML

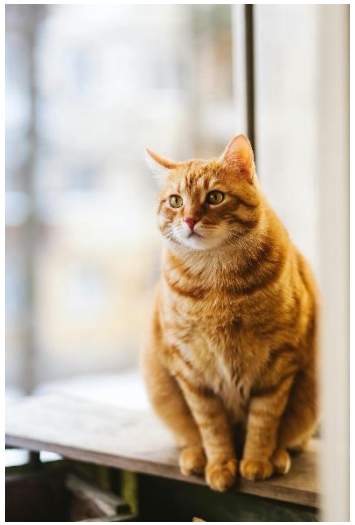
- Weakly-supervised Learning
- Semi-supervised Learning
- Self-supervised Learning

Hypothesis Space

Inductive Learning (for Supervised Setting)

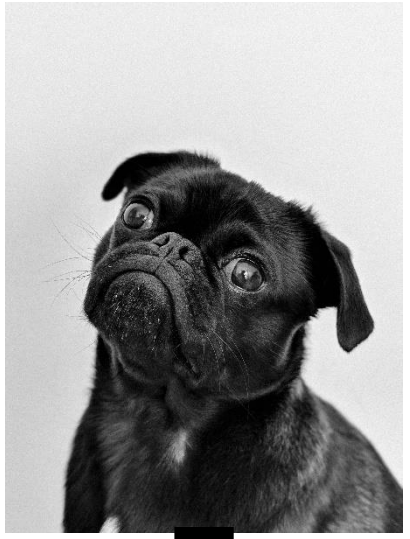
- Given data X and Label Y
  - Or, we may consider data X and a function $f(X)$
    - $f(X)$: Categorical for classification
    - $f(X)$: Real number for regression
    - $f(X)$: Probability of X for probability estimation
- Learning from experience

Inductive Learning



$f(\cdot)$

Cat



$f(\cdot)$

Dog

https://www.pexels.com/

Inductive Learning

$f(\cdot)$

1.7 Cr

https://www.pexels.com/

Inductive Learning (for Supervised Setting)

- Given data X and Label Y
  - Or, we may consider data X and a function $f(X)$
    - $f(X)$: Categorical for classification
    - $f(X)$: Real number for regression
    - $f(X)$: Probability of X for probability estimation

- Learning from experience
  - **Given some data, we are performing an induction to identify the function $\boldsymbol{f}(\cdot)$**

Inductive Learning

- Necessary because we just have sample of data and not the entire population

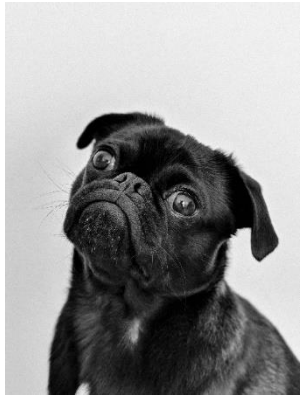
Inductive Learning (for Supervised Setting)

- Given data X and Label Y
  - Or, we may consider data X and a function $f(X)$

- Return a function h that approximates f
  - **We don't know f**
  - **But we assume that there is an f**
  - **h is called a hypothesis**

The Data

- Data X and Label Y



- For X
  - We may use raw data, such as an image
  - Or, we may use features, such as height, weight, colour, etc. of the object

Feature

- Characteristics/ properties that describe an instance of data
  - Height, weight, etc.
  - Sometime, these are called attributes as well
- Feature vector
  - A vector with features as dimension

(a) (b) (c)

Figure 1.1: Three types of Iris flowers: Setosa, Versicolor and Virginica. Used with kind permission of Dennis Kramb and SIGNA.

index	sl	sw	pl	pw	label
0	5.1	3.5	1.4	0.2	Setosa
1	4.9	3.0	1.4	0.2	Setosa
	...				
50	7.0	3.2	4.7	1.4	Versicolor
	...				
149	5.9	3.0	5.1	1.8	Virginica

Table 1.1: A subset of the Iris design matrix. The features are: sepal length, sepal width, petal length, petal width. There are 50 examples of each class.

How to Obtain Features?

- May be obtained through measurement at the time of data collection
  - Such as height, weight, etc. of different animals
- May be computed from raw data
  - Such as texture, SIFT features, etc.
- Feature engineering
- Many neural network methods can directly deal with raw data without explicitly requiring features at the input

Feature Space

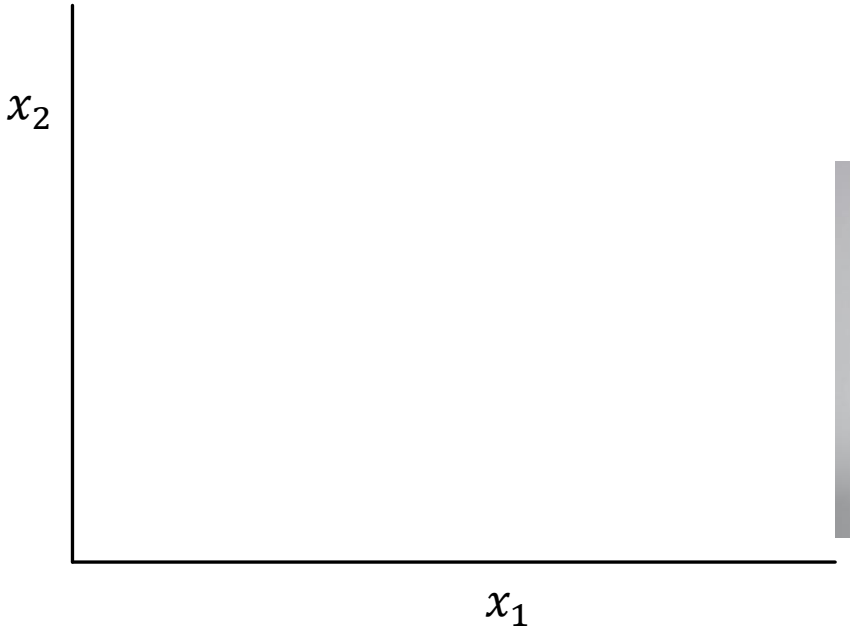
CAT

DOG

https://www.pexels.com/

Feature Space

CAT



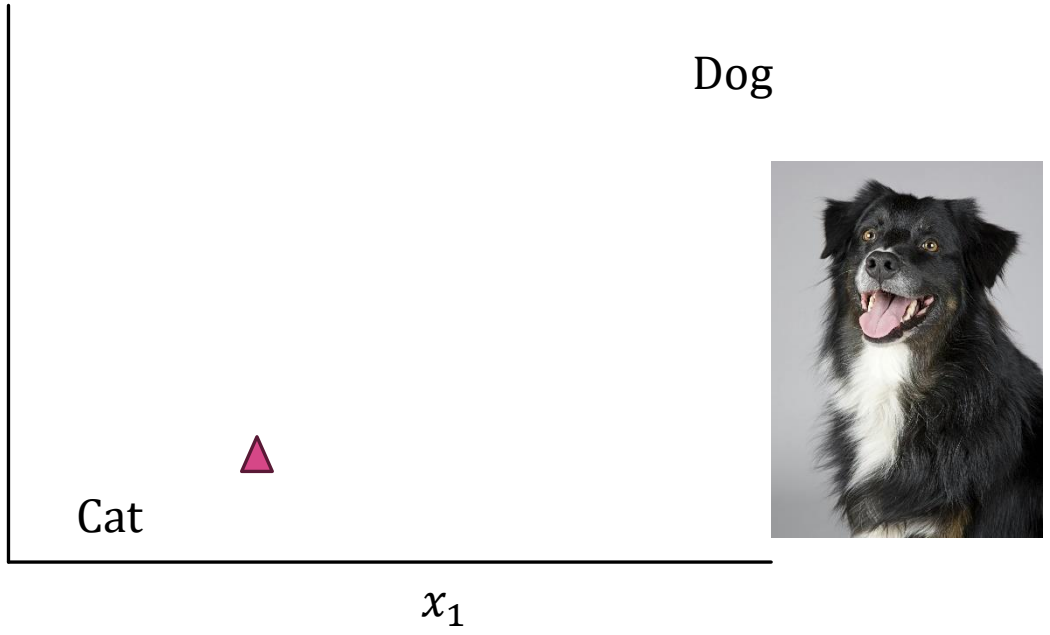
DOG

x_1: Height
x_2: Weight

https://www.pexels.com/

Feature Space

CAT



DOG

x_1: Height
x_2: Weight

https://www.pexels.com/

Feature Space

CAT

x_2

Dog

Cat

x_1

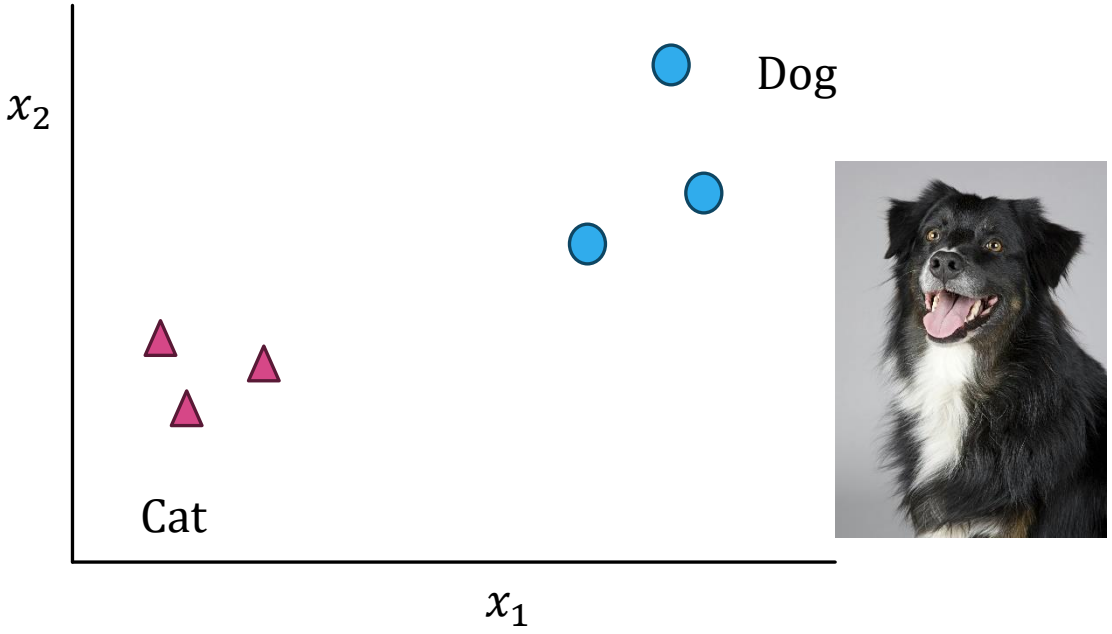
DOG

x_1: Height
x_2: Weight

https://www.pexels.com/

Feature Space

CAT

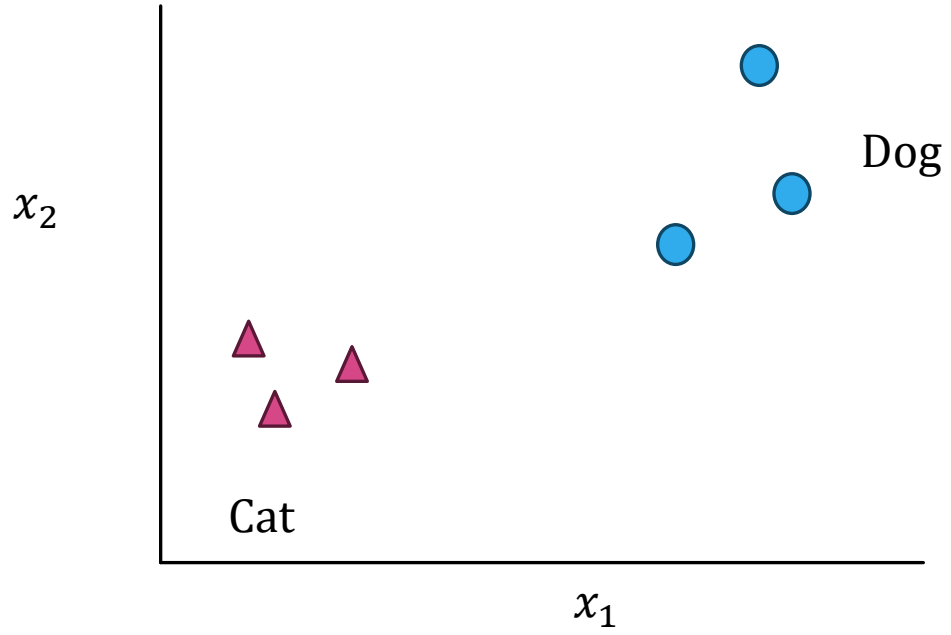


DOG

x_1: Height
x_2: Weight

https://www.pexels.com/

Feature Space



The space defined by features is called feature space

Each instance is a point in the feature space

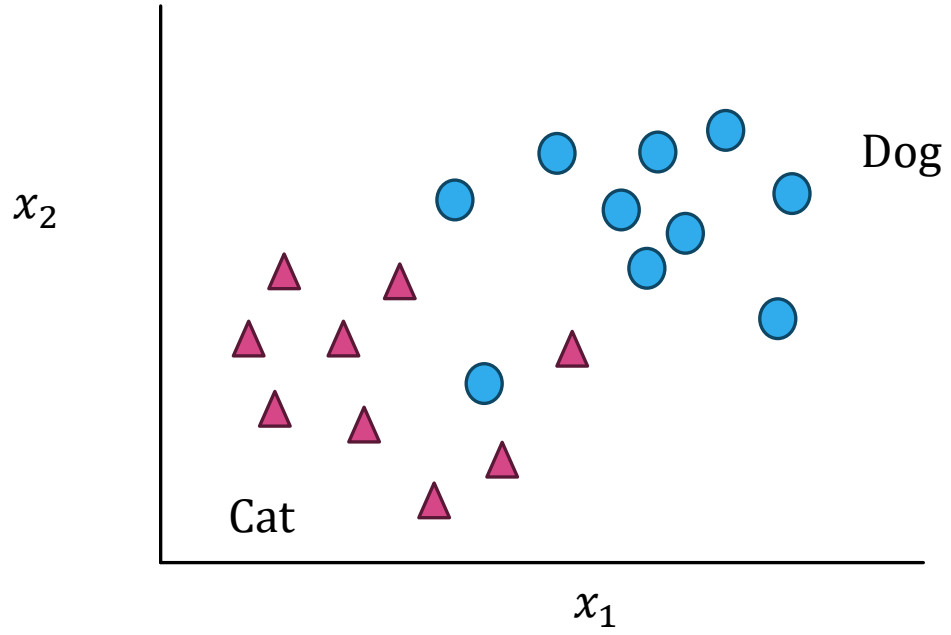
x_1: Height
x_2: Weight

Inductive Learning (for Supervised Setting)

- Given data X and Label Y
  - Or, we may consider data X and a function $f(X)$

- Return a function h that approximates f
  - **We don't know f**
  - **But we assume that there is an f**
  - **h is called hypothesis**

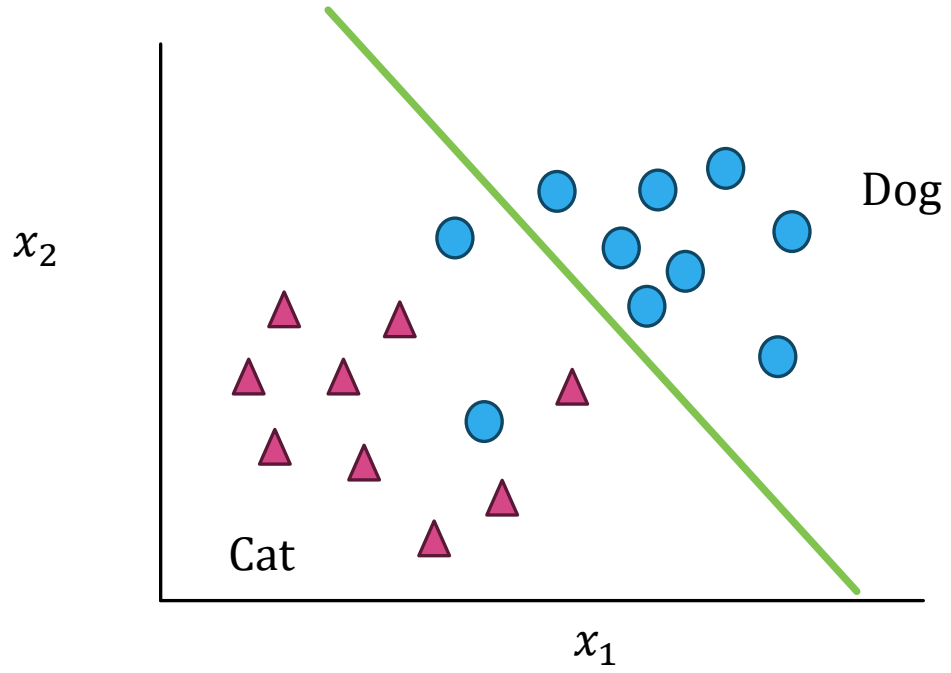
Hypothesis



Suppose, I want a function that can differentiate cats and dogs

x_1: Height
x_2: Weight

Hypothesis

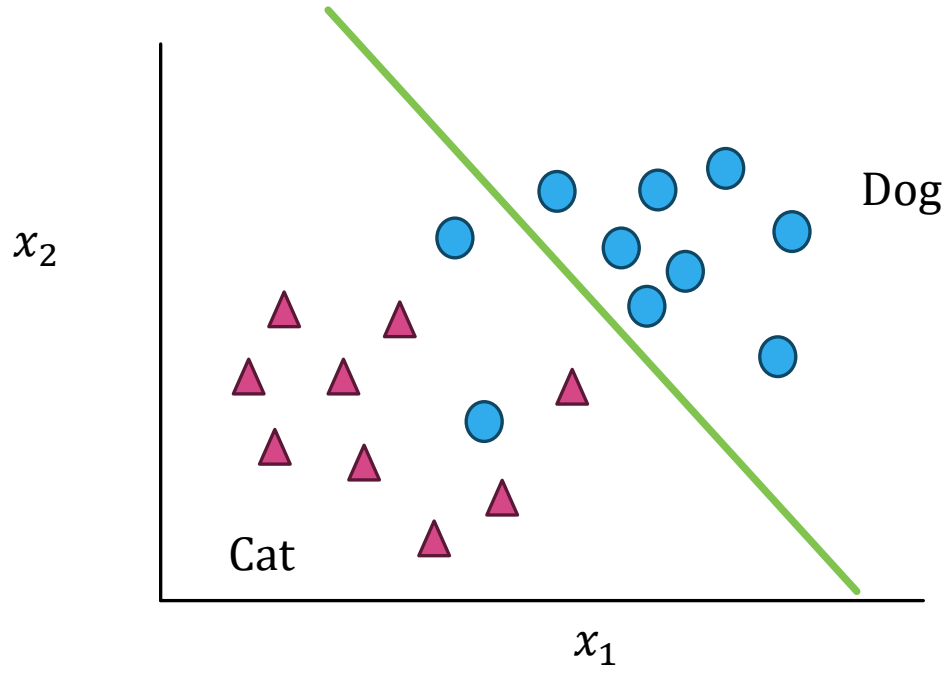


Suppose, I want a function that can differentiate cats and dogs

The green straight line is such a function

x_1: Height
x_2: Weight

Hypothesis



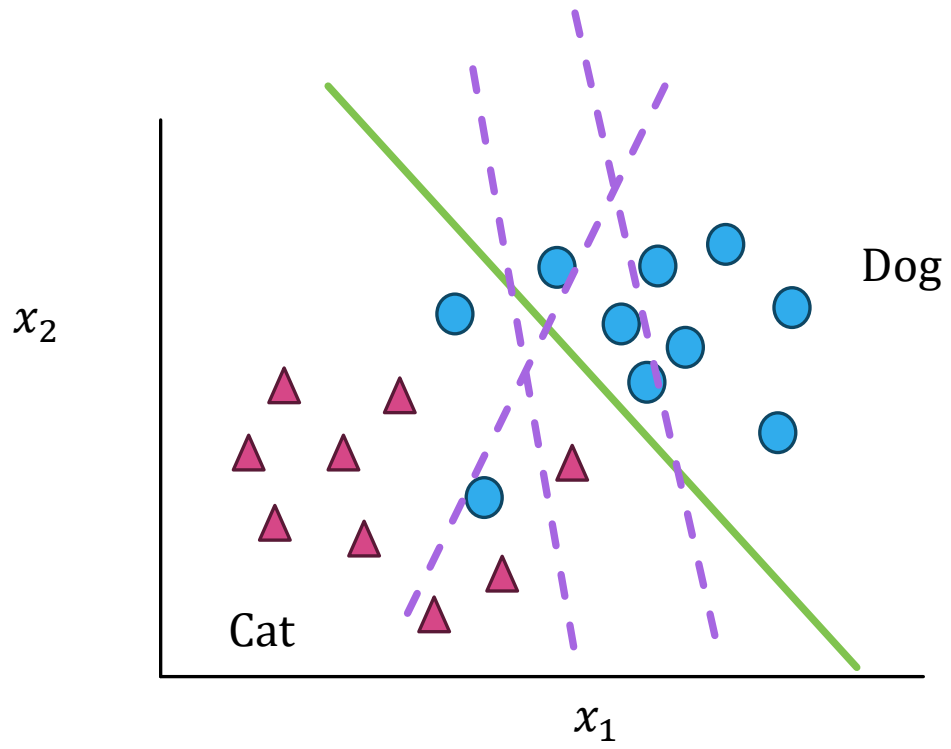
x_1: Height
x_2: Weight

Suppose, I want a function that can differentiate cats and dogs

The green straight line is such a function

The function is called a hypothesis

Hypothesis



x_1: Height
x_2: Weight

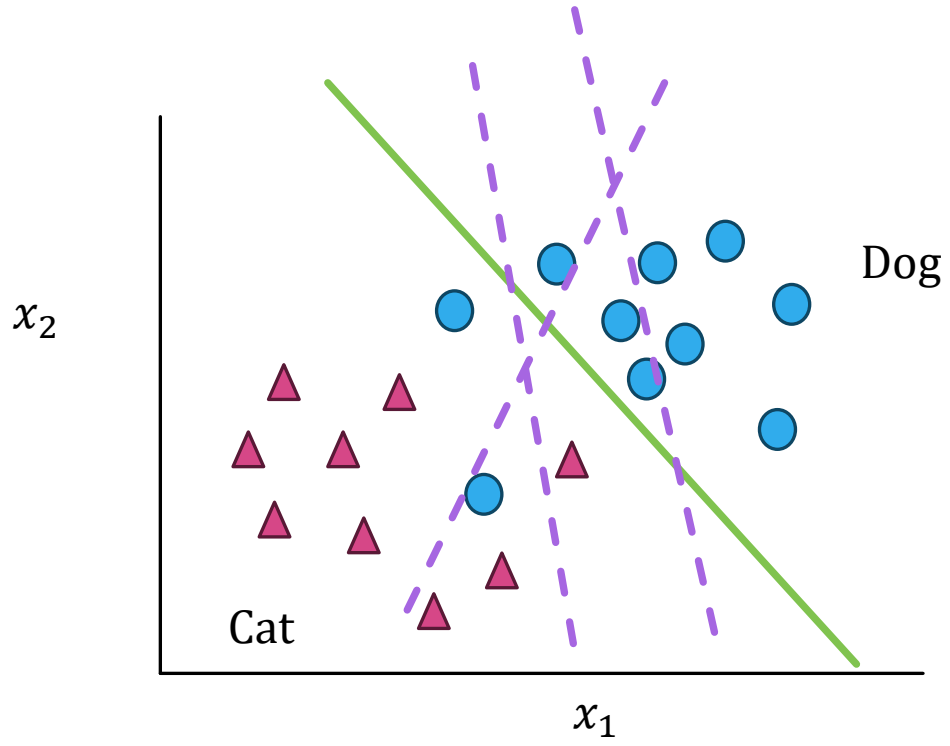
Suppose, I want a function that can differentiate cats and dogs

The green straight line is such a function

The function is called a hypothesis

Many such hypothesis are possible

Hypothesis



x_1: Height
x_2: Weight

Suppose, I want a function that can differentiate cats and dogs

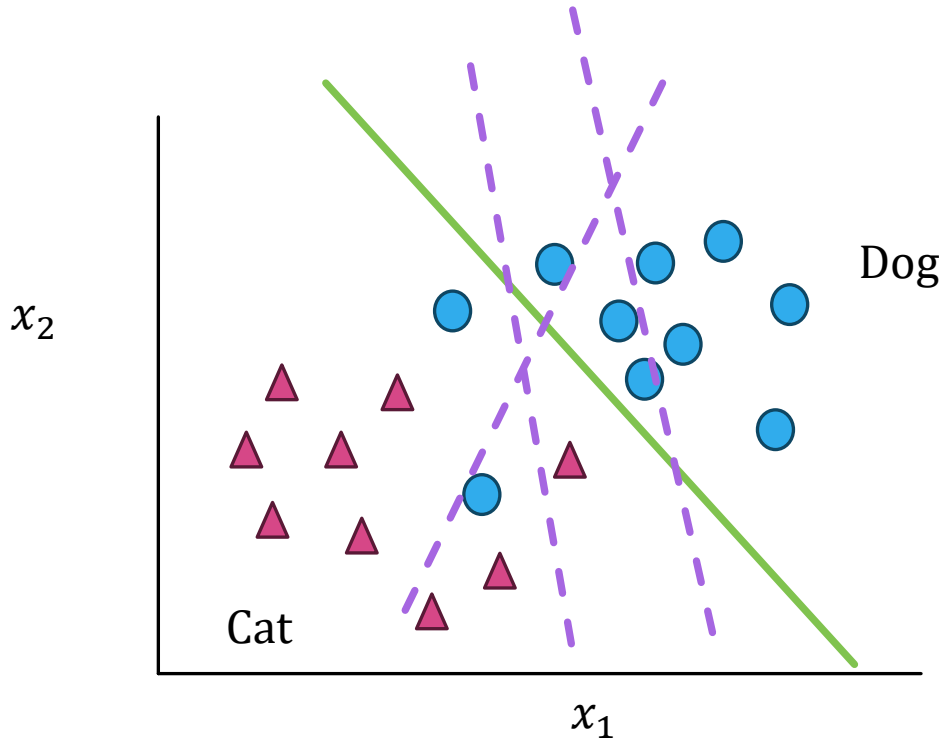
The green straight line is such a function

It's called a hypothesis

Many such hypothesis are possible

Hypothesis space: set of all such legal hypotheses

Hypothesis

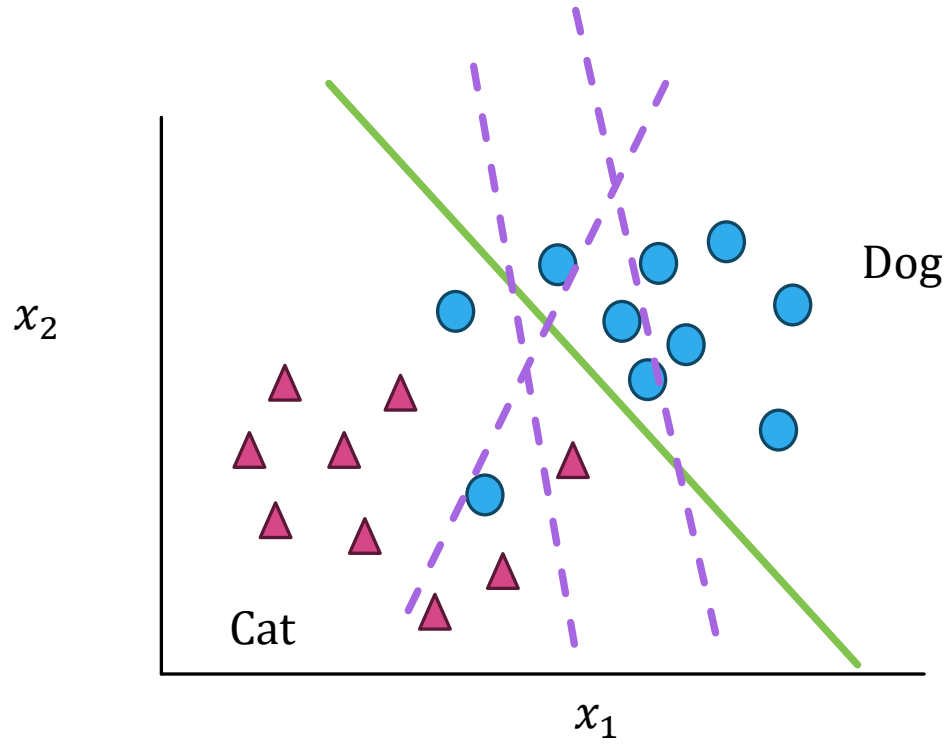


So, for a problem, we first define the hypothesis space

Then, we find the best hypothesis given the data

x_1: Height
x_2: Weight

Hypothesis



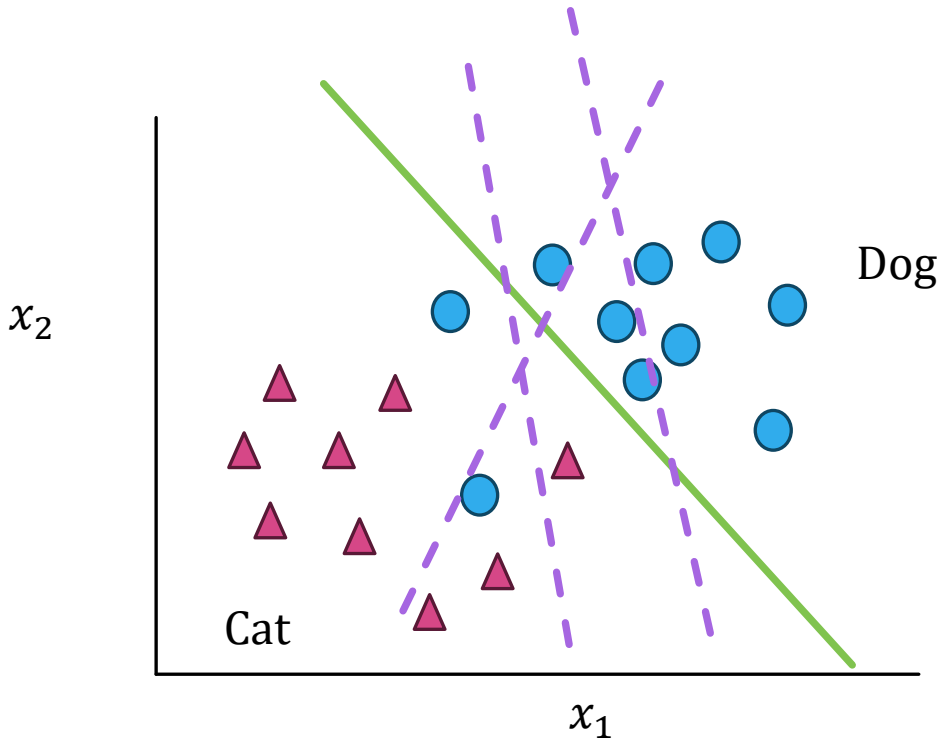
So, for a problem, we first define the hypothesis space

Then, we find the best hypothesis given the data

But, how many such hypotheses are possible?

x_1: Height
x_2: Weight

Hypothesis



x_1: Height
x_2: Weight

But, how many such hypotheses are possible?

Let's assume that I have N data points

Each data point can be classified in one of the two classes

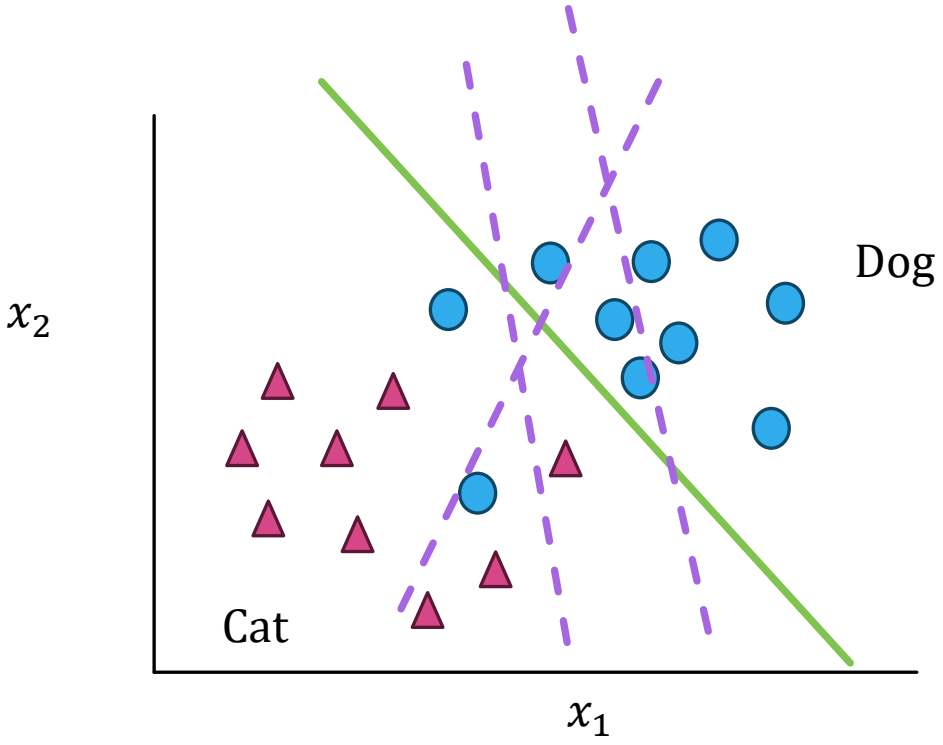
So, possible class combination for N data points is 2^N

One hypothesis for each class combination

So, 2^N hypothesis are possible

It could be very very large

Inductive Bias



So, 2^N hypothesis are possible

It could be very very large

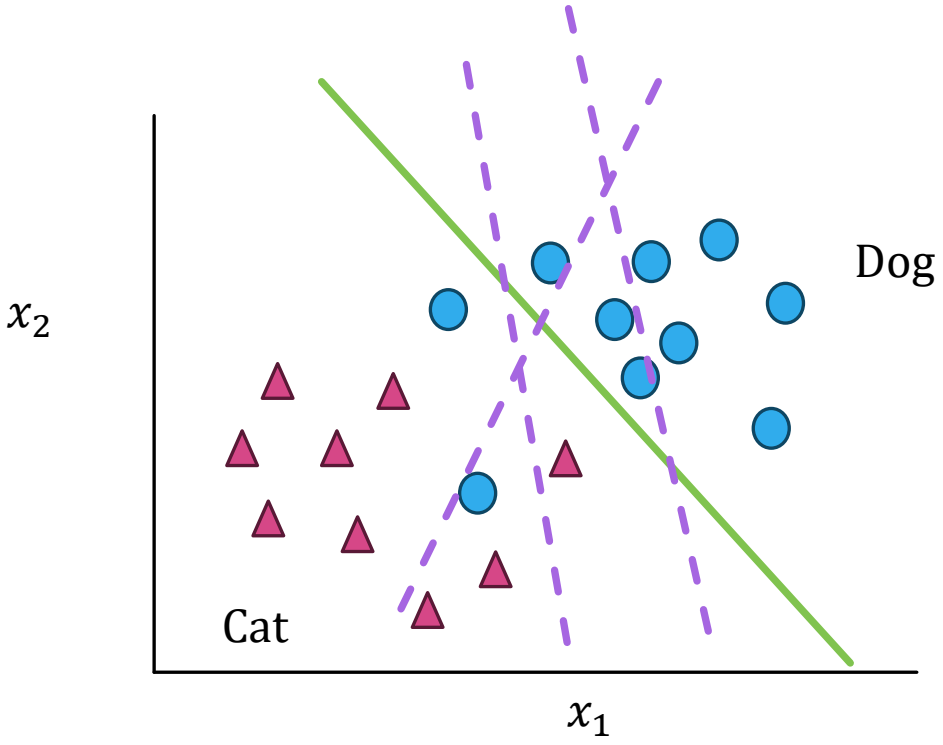
We can't practically find the best hypothesis from such a large set

So, we have to make some restrictive assumption

Inductive Bias

x_1: Height
x_2: Weight

Inductive Bias



So, 2^N hypothesis are possible
It could be very very large

We can't practically find the best hypothesis from such a large set

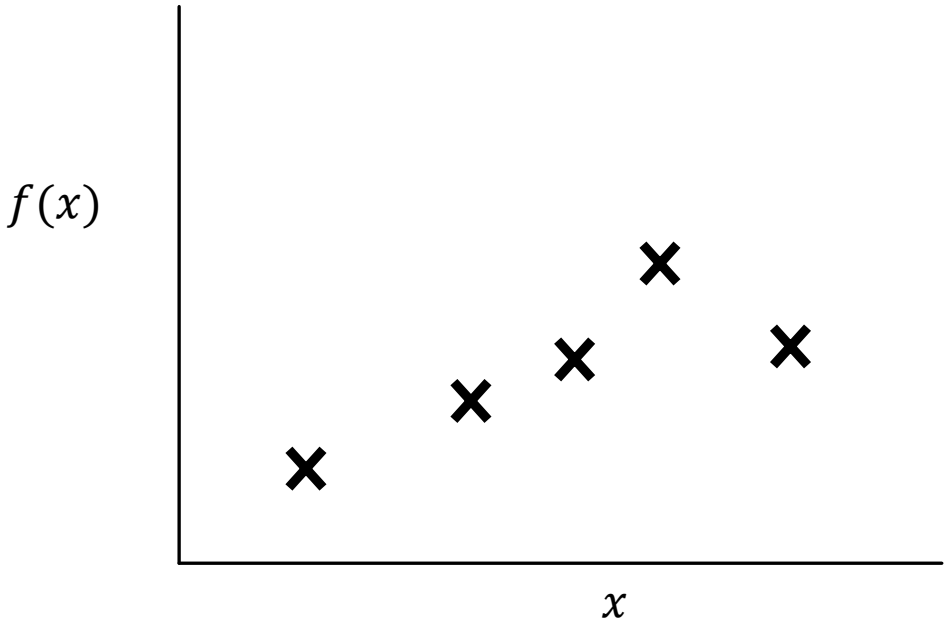
So, we have to make some restrictive assumption

Inductive Bias

So, we select some hypothesis language

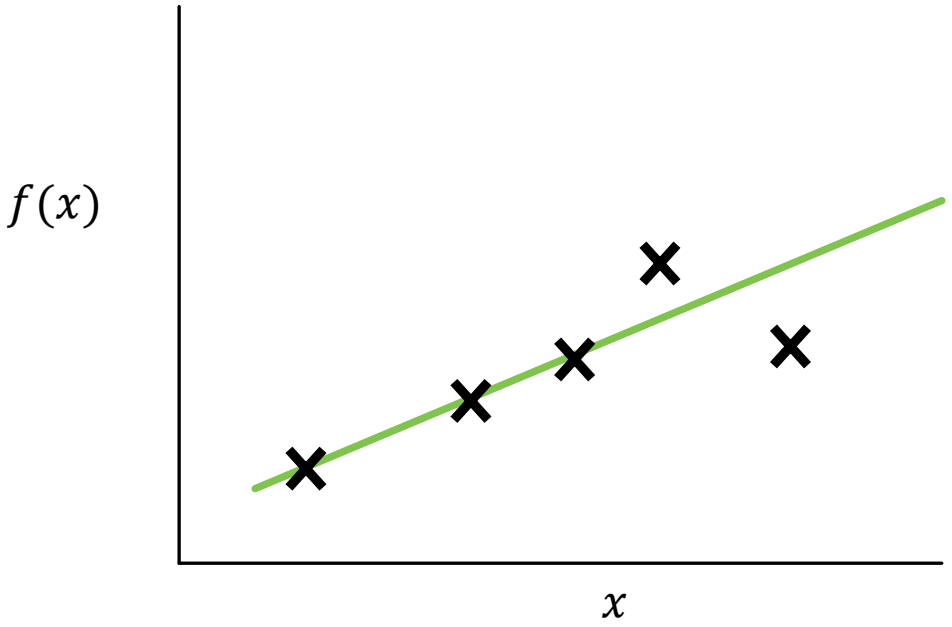
x_1: Height
x_2: Weight

Inductive Bias



Consider a regression problem with data samples x and regression value $f(x)$

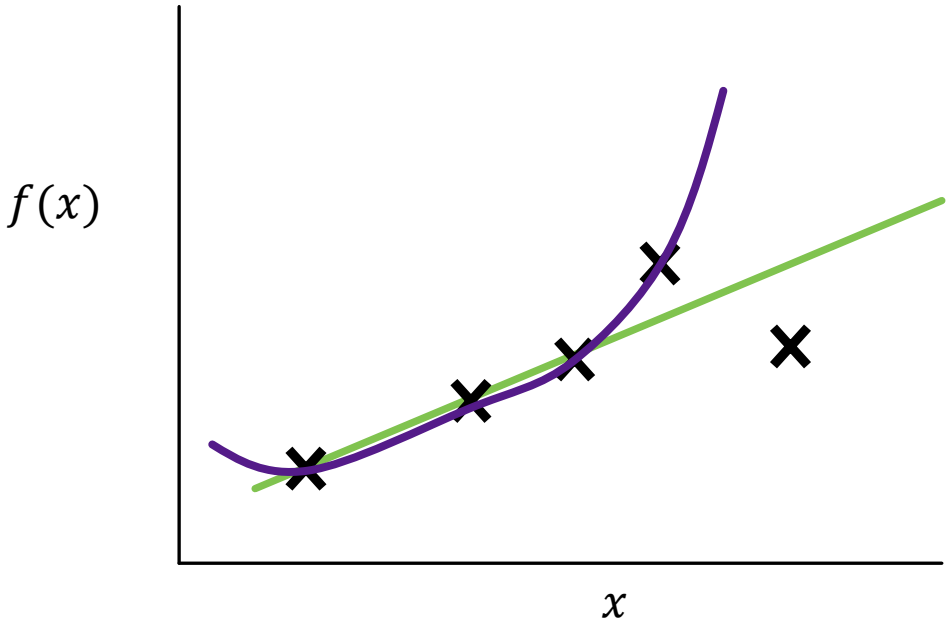
Hypothesis



Consider a regression problem with data samples x and regression value $f(x)$

Which curve fits the best?

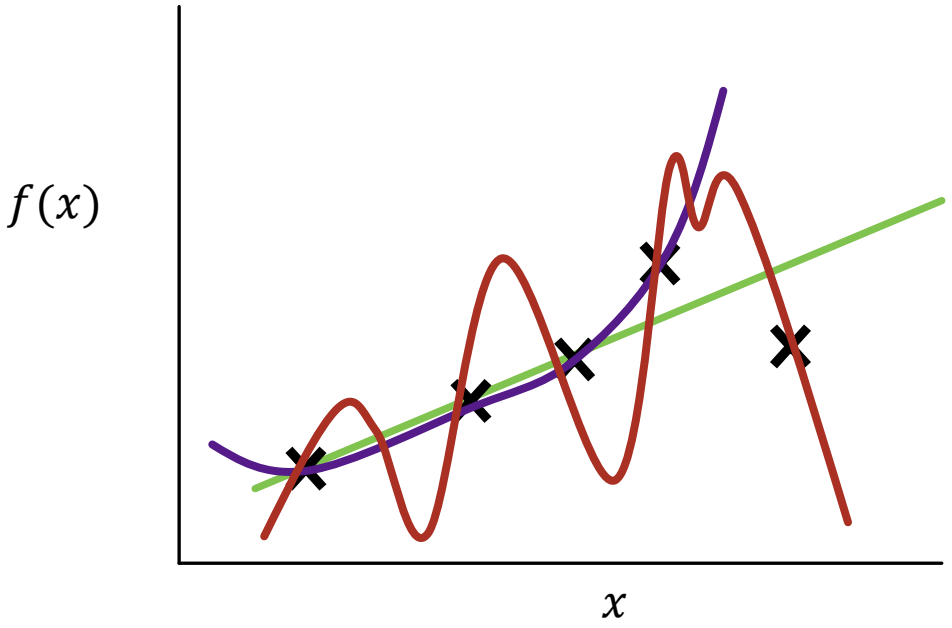
Hypothesis



Consider a regression problem with data samples x and regression value $f(x)$

Which curve fits the best?

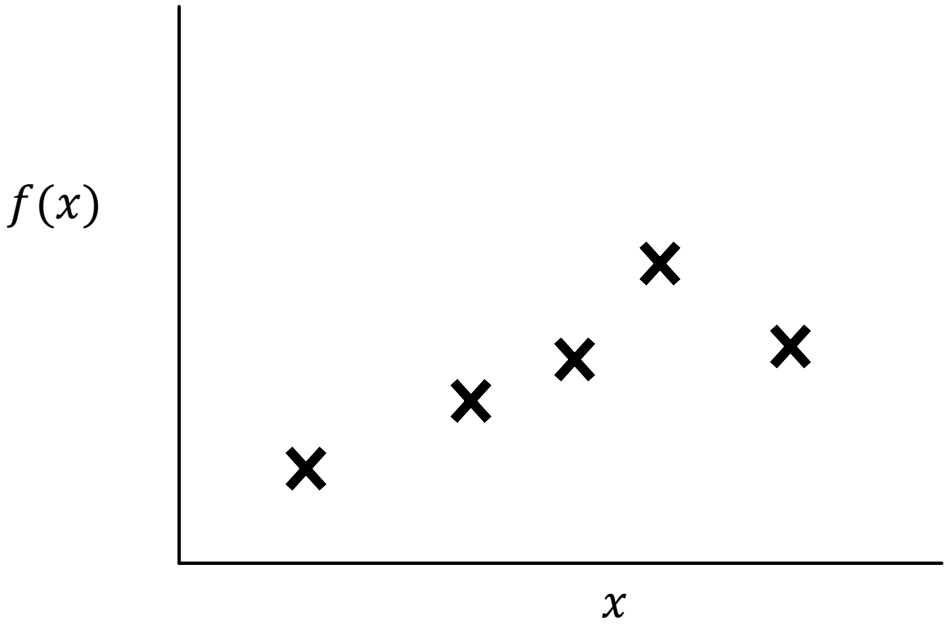
Hypothesis



Consider a regression problem with data samples x and regression value $f(x)$

Which curve fits the best?

Hypothesis

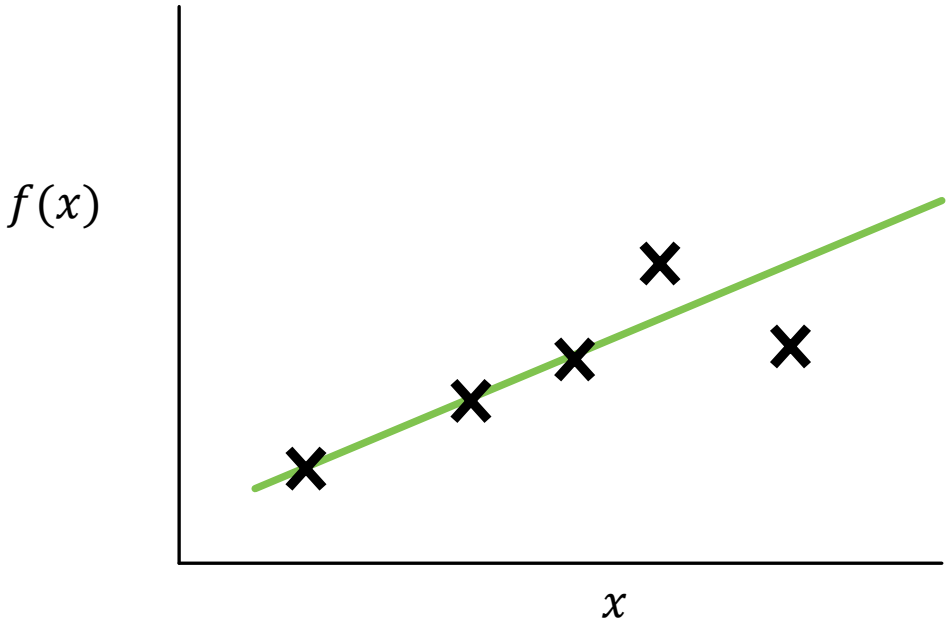


Consider a regression problem with data samples x and regression value $f(x)$

Which curve fits the best?

What if I tell you
x is current and $f(x)$ is voltage?

Hypothesis



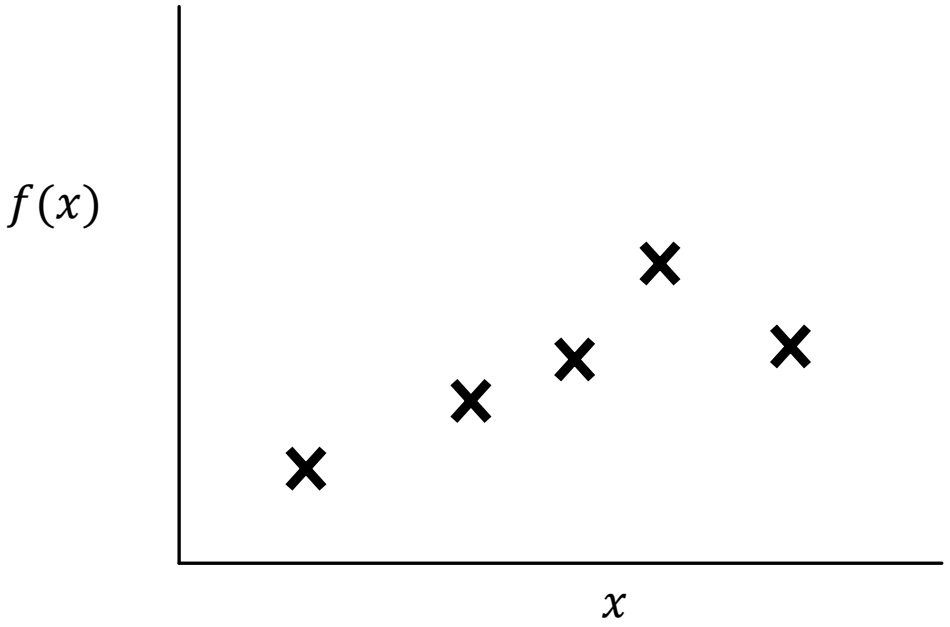
Consider a regression problem with data samples x and regression value $f(x)$

Which curve fits the best?

What if I tell you
x is current and $f(x)$ is voltage?

Probably the green curve fits the best and the points which do not lie on the green curve may be considered as outlier

Hypothesis

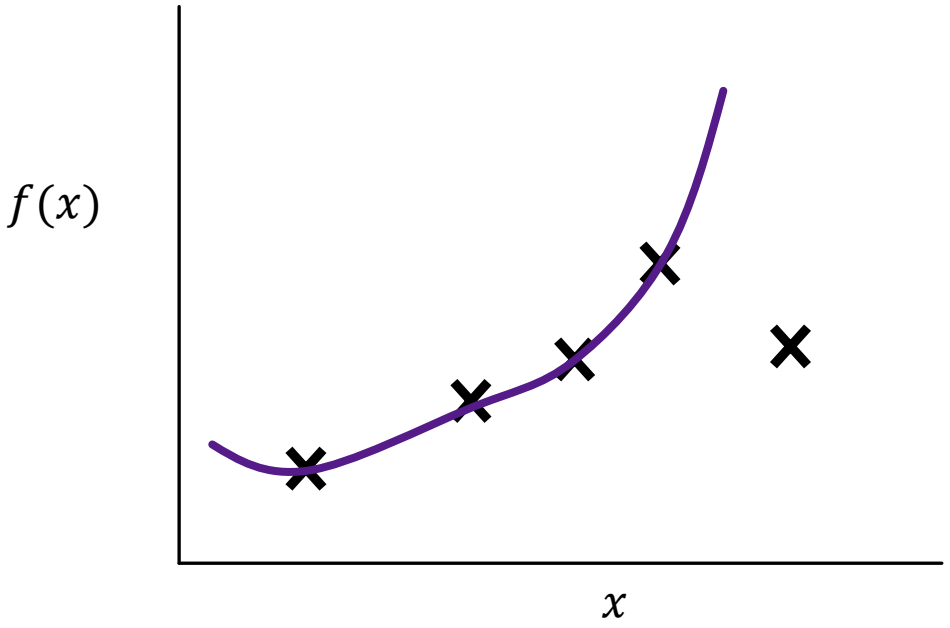


Consider a regression problem with data samples x and regression value $f(x)$

Which curve fits the best?

What if I tell you
x is time of the day and $f(x)$ is temperature?

Hypothesis



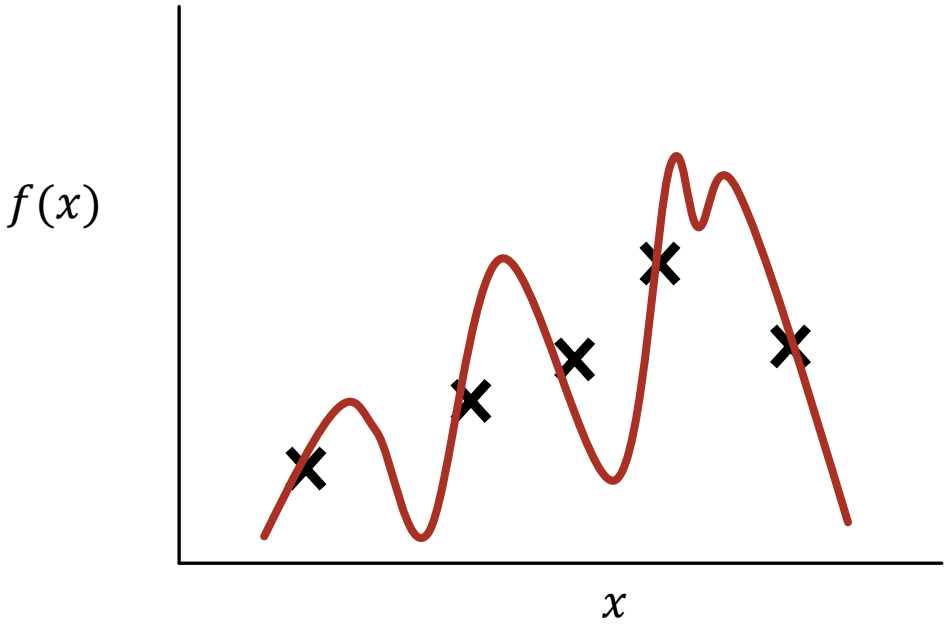
Consider a regression problem with data samples x and regression value $f(x)$

Which curve fits the best?

What if I tell you
x is time of the day and $f(x)$ is temperature?

Probably the purple curve fits the best and the points which do not lie on the purple curve may be considered as outlier

Hypothesis

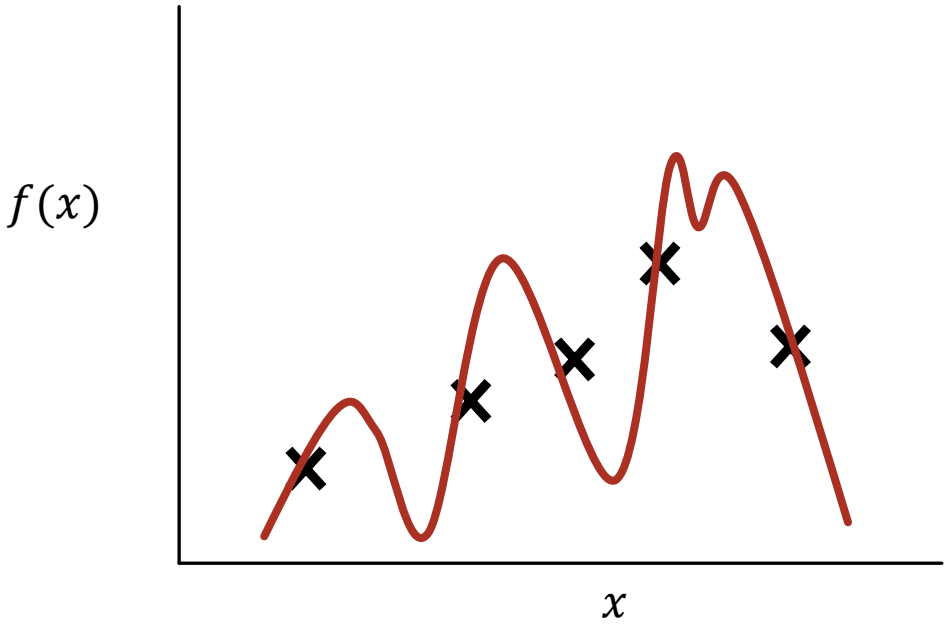


Consider a regression problem with data samples x and regression value $f(x)$

Which curve fits the best?

What if I tell you
x is day and $f(x)$ is the price of a stock?

Hypothesis



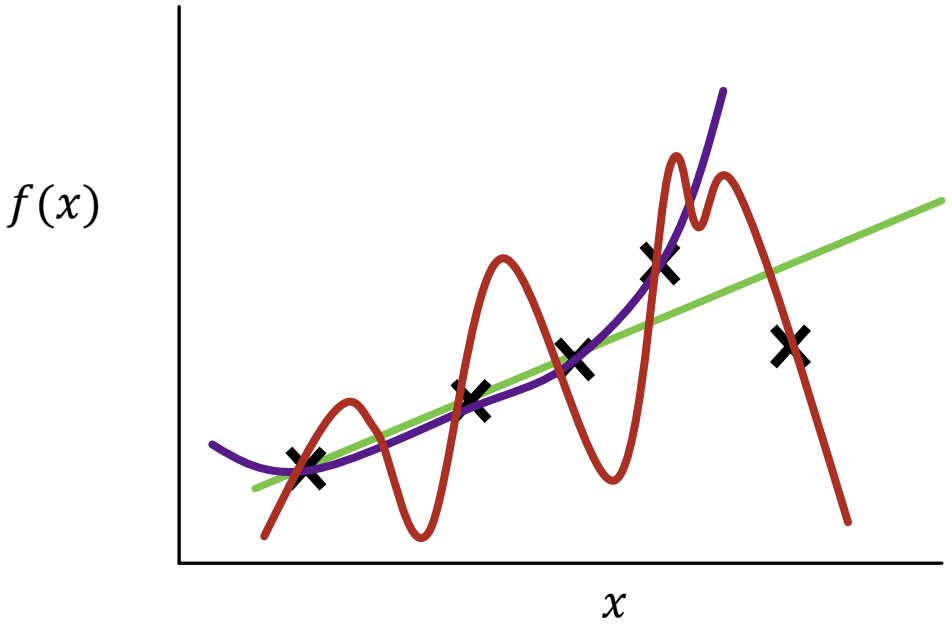
Consider a regression problem with data samples x and regression value $f(x)$

Which curve fits the best?

What if I tell you
x is day and $f(x)$ is the price of a stock?

Probably the red curve fits the best and the points which do not lie on the red curve may be considered as outlier

Hypothesis



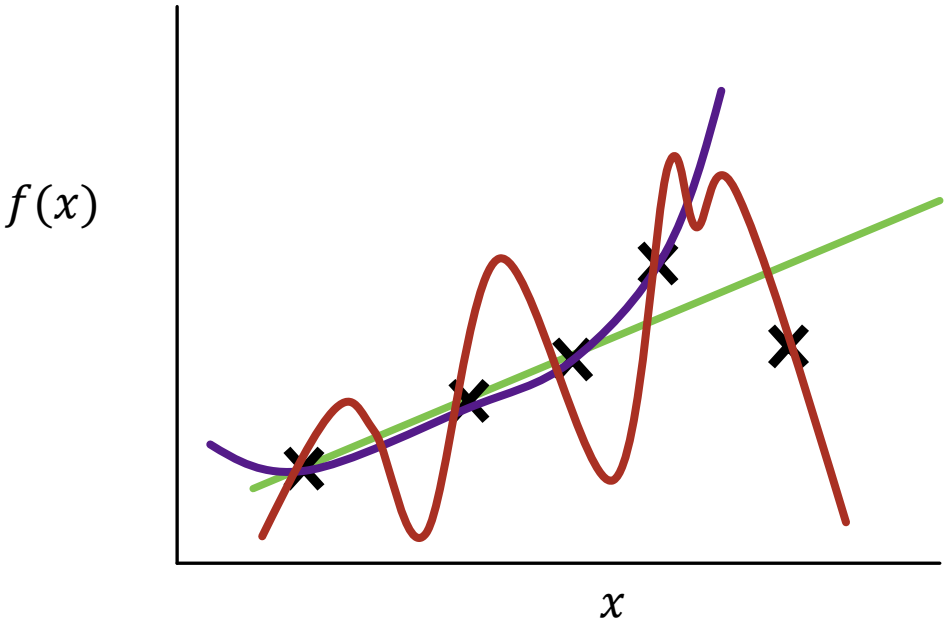
Consider a regression problem with data samples x and regression value $f(x)$

Which curve fits the best?

That means our decision of fitting the curve depends on our assumption about the problem

We may consider that the samples can be explained by a straight line, or by a polynomial curve, or may be a more complex curve

Inductive Bias



Consider a regression problem with data samples x and regression value $f(x)$

Which curve fits the best?

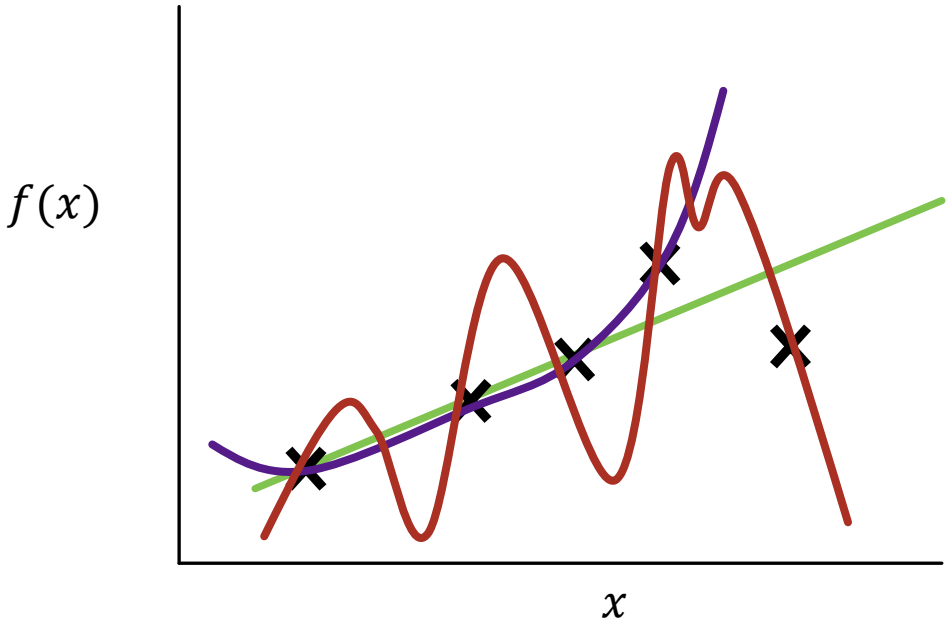
That means our decision of fitting the curve depends on our assumption about the problem

We may consider that the samples can be explained by a straight line, or by a polynomial curve, or may be a more complex curve

Inductive Bias

No Free Lunch theorem

Generalization

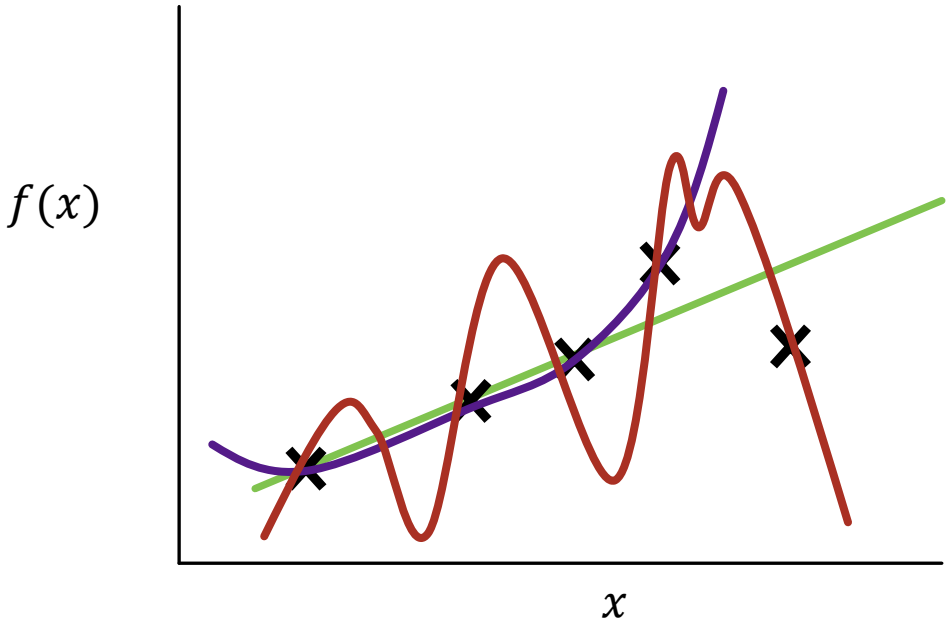


We want to find a curve (hypothesis) that will generalize well (fits the unseen examples in accurately)

Ockham's razor: prefer the simplest hypothesis consistent with data

This is just an assumption and does not always work

Generalization



We want to find a curve (hypothesis) that will generalize well (fits the unseen examples in accurately)

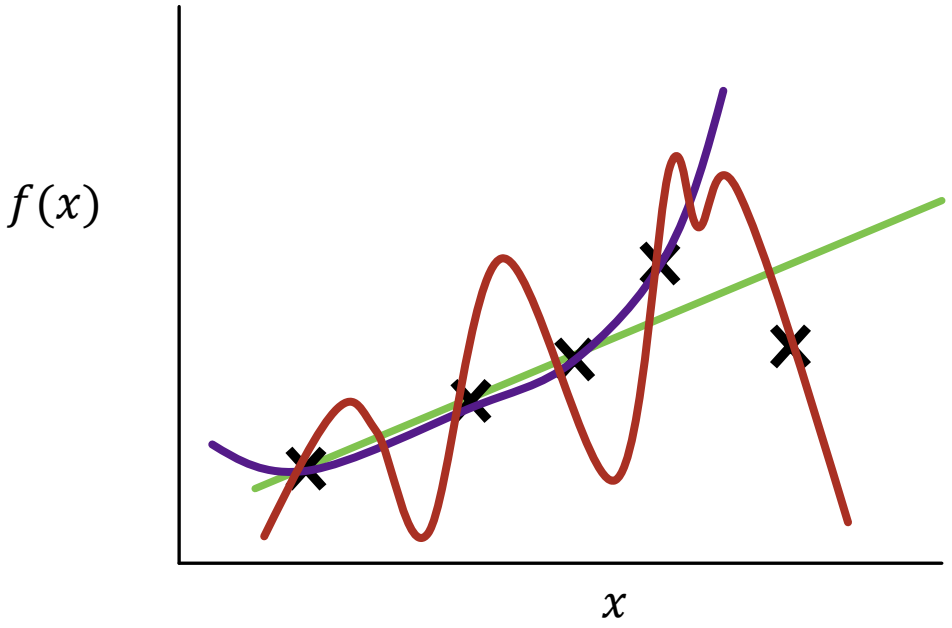
Ockham's razor: prefer the simplest hypothesis consistent with data

This is just an assumption and does not always work

In many occasions, nature follows simple laws and simple states

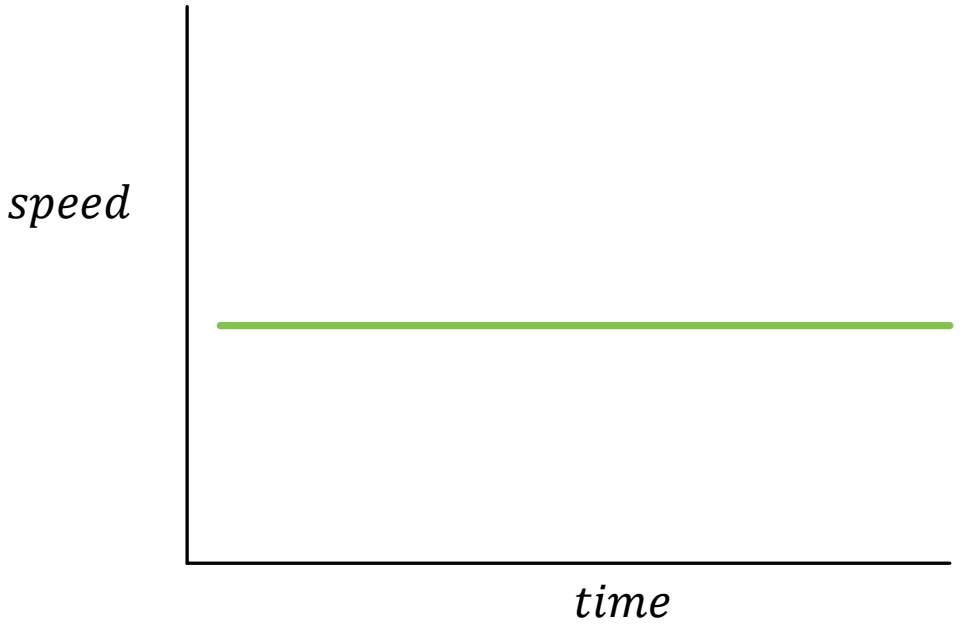
So, if we come up with a simple hypothesis, it may generalize well

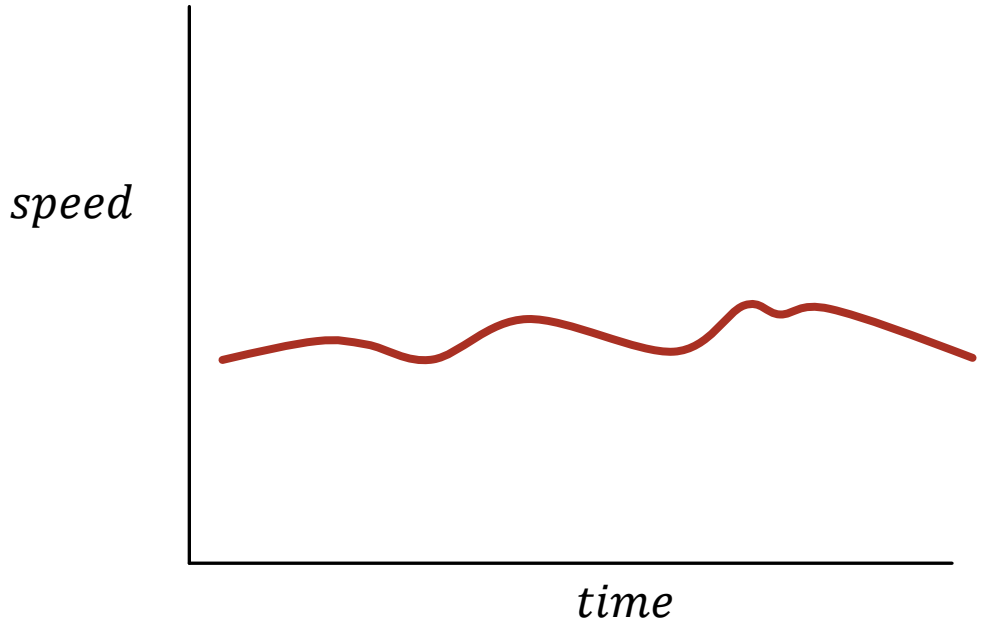
Inductive Bias



- More assumptions: more bias
- Which curve has more bias?

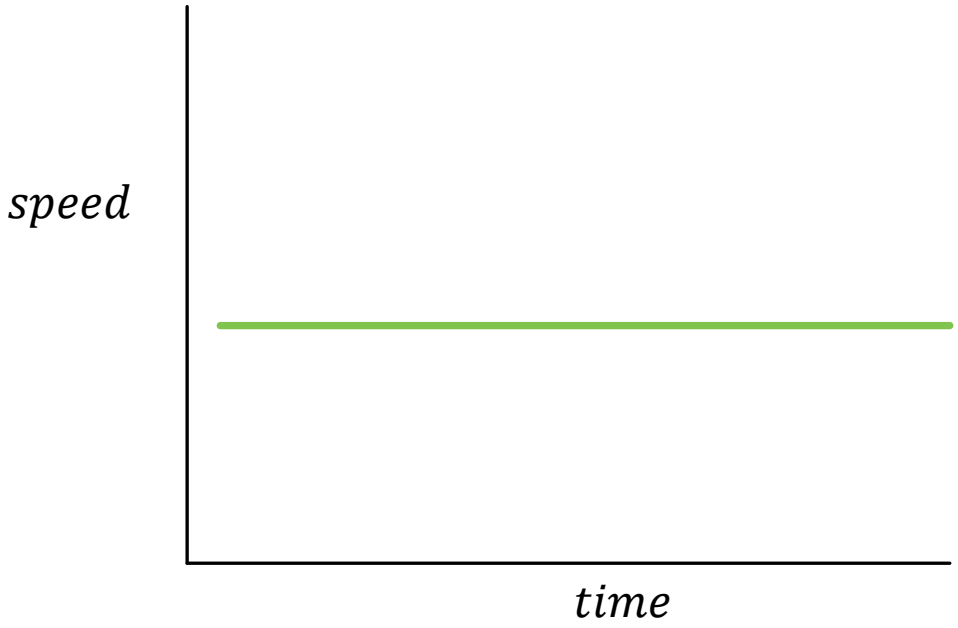
Assumption and Complexity of the Curve (Model)

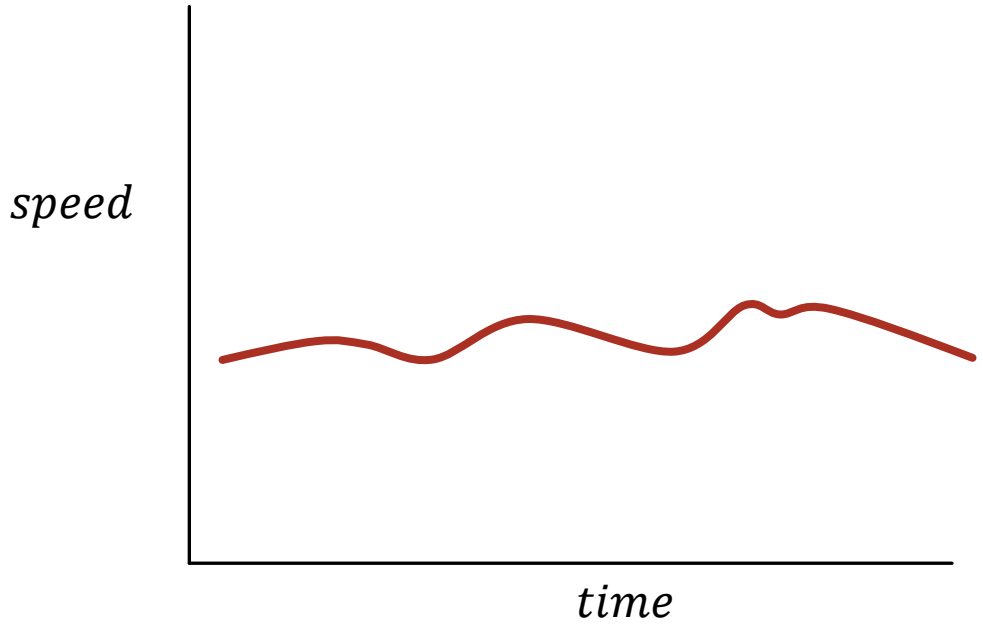




- Suppose, I plot the speed of a car over time
- Which of the curves is more realistic?

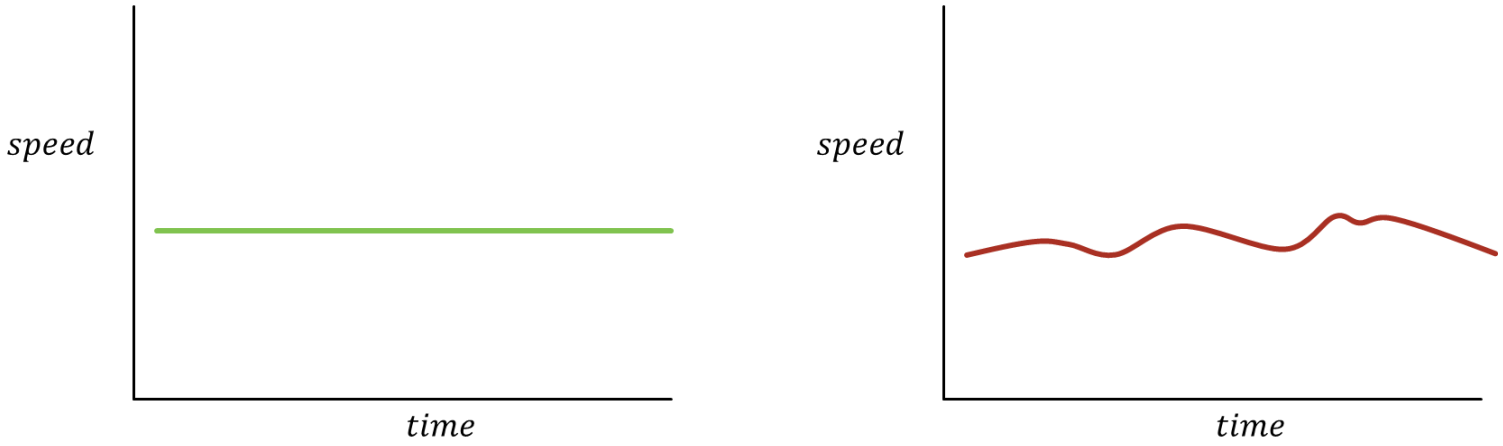
Assumption and Complexity of the Curve (Model)





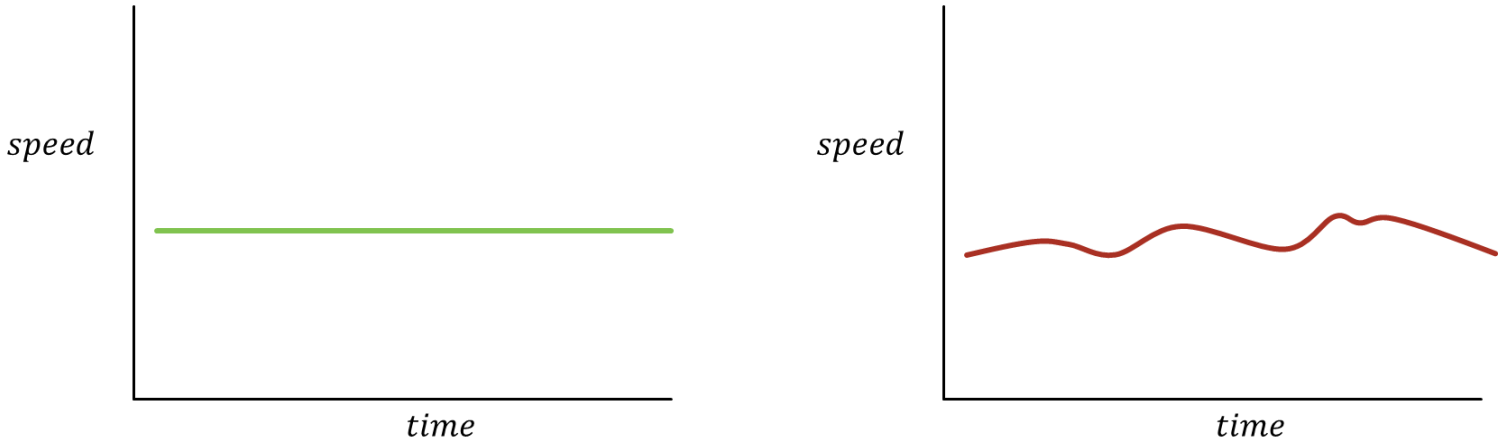
- Suppose, I plot the speed of a car over time
- Which of the curves is more realistic?
  - The red curve

Assumption and Complexity of the Curve (Model)



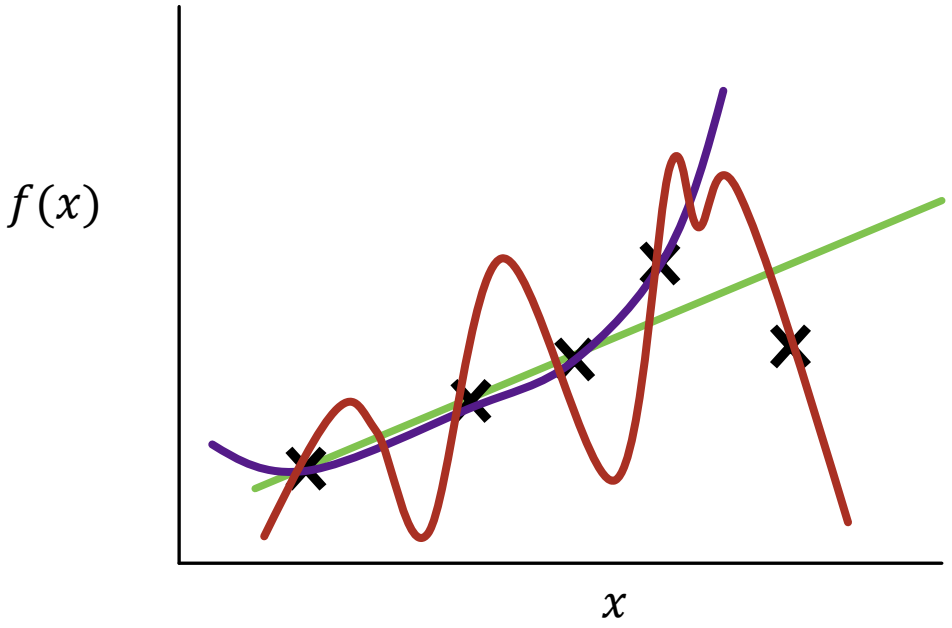
- Suppose, I plot the speed of a car over time
- Which of the curves is more realistic?
  - The red curve
- More realistic curve -> Less assumption -> More complex curve
- For the green curve to be true, we have to make a lot of assumptions
  - e.g., the road is smooth, very less traffic, almost no braking, no signal, etc.

Assumption and Complexity of the Curve (Model)



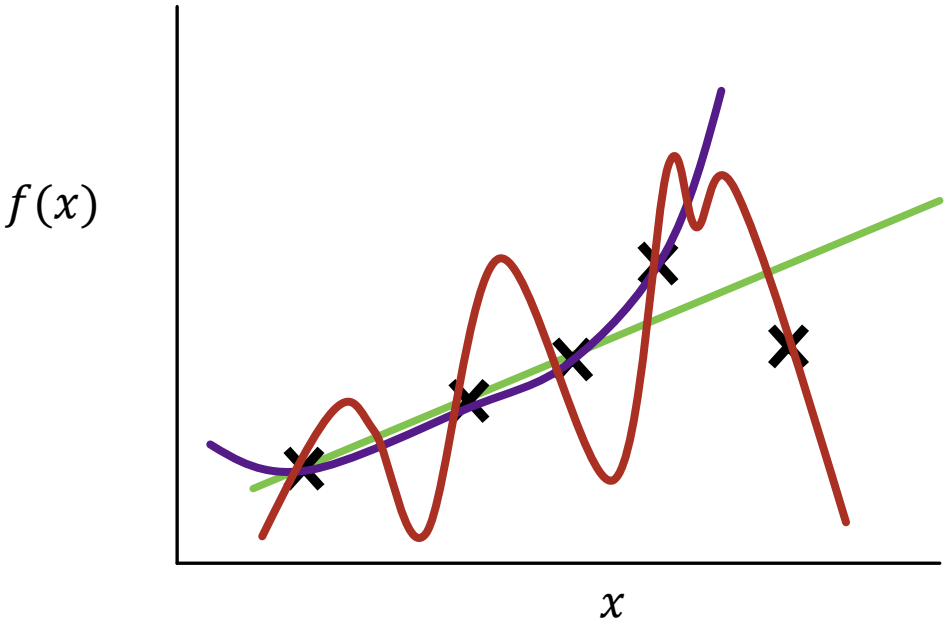
- Which of the curves is more realistic?
  - The red curve
- More realistic curve -> Less assumption -> More complex curve
- For the green curve to be true, we have to make a lot of assumptions
  - e.g., the road is smooth, very less traffic, almost no braking, no signal, etc.
- So, simple curves (models) require more assumption -> more bias

Inductive Bias



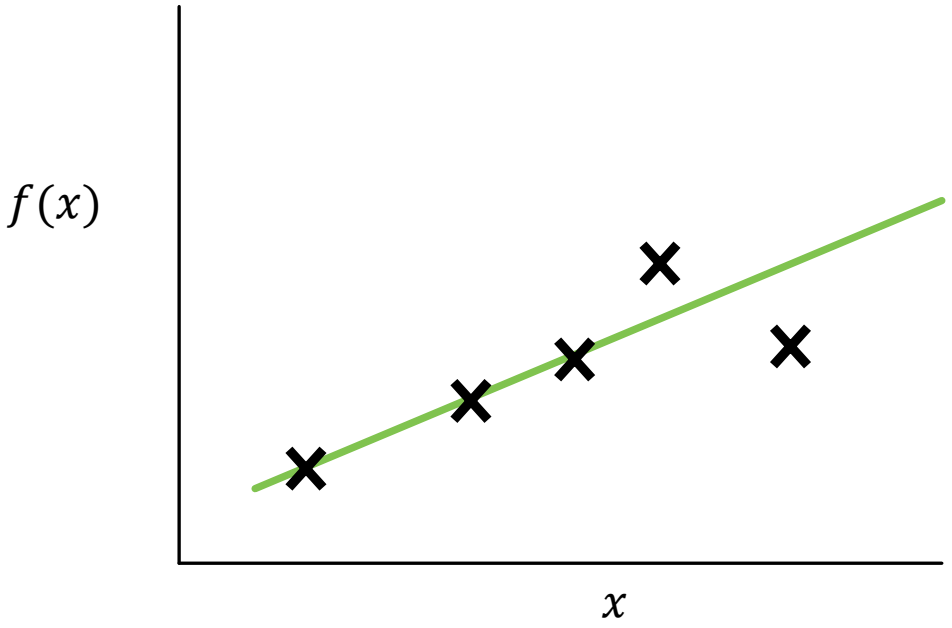
- More assumptions: more bias
- More assumption: simple model/ simple curve
  - More bias

Types of Inductive Bias



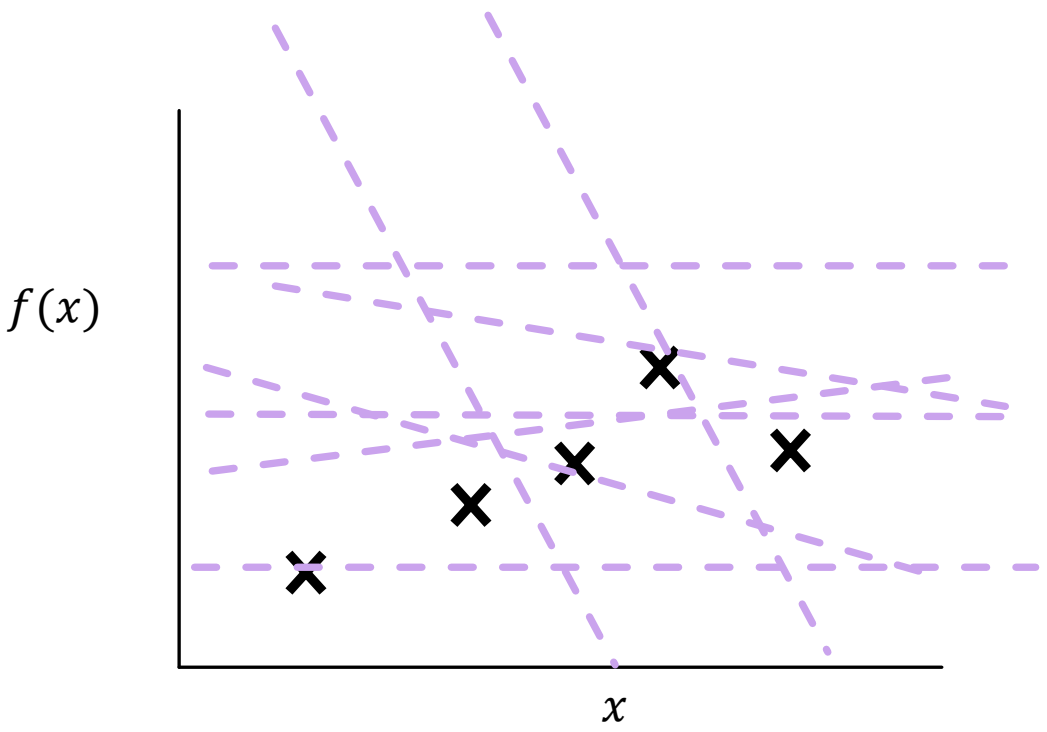
- Restriction bias (aka Absolute bias)
  - Representational power of algorithm
  - The set of hypothesis that an algorithm will consider
    - E.g., an algorithm may consider a hypotheses of straight lines
- Preference bias (aka Relative bias)
  - Indicates what representation(s) a learning algorithm prefers
  - For example, an algorithm may prefer straight lines parallel to x-axis

Types of Inductive Bias



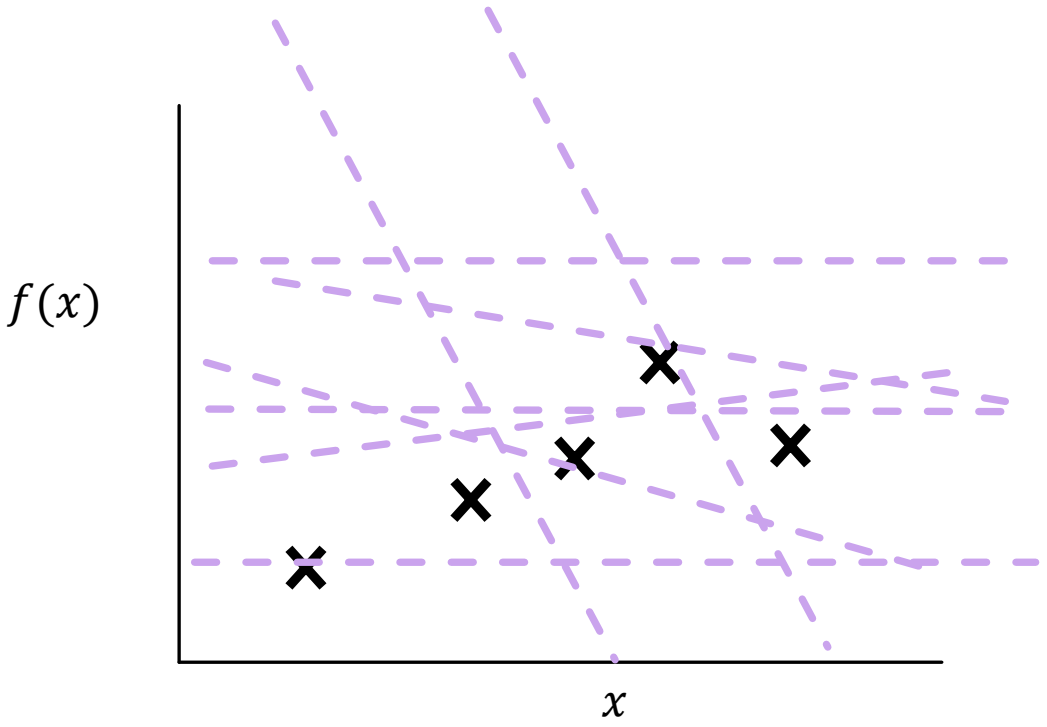
- Restriction bias (aka Absolute bias)
  - Representational power of algorithm
  - The set of hypothesis that an algorithm will consider
    - E.g., an algorithm may consider a hypotheses of straight lines
- Preference bias (aka Relative bias)
  - Indicates what representation(s) a learning algorithm prefers
  - For example, an algorithm may prefer straight lines parallel to x-axis

Types of Inductive Bias



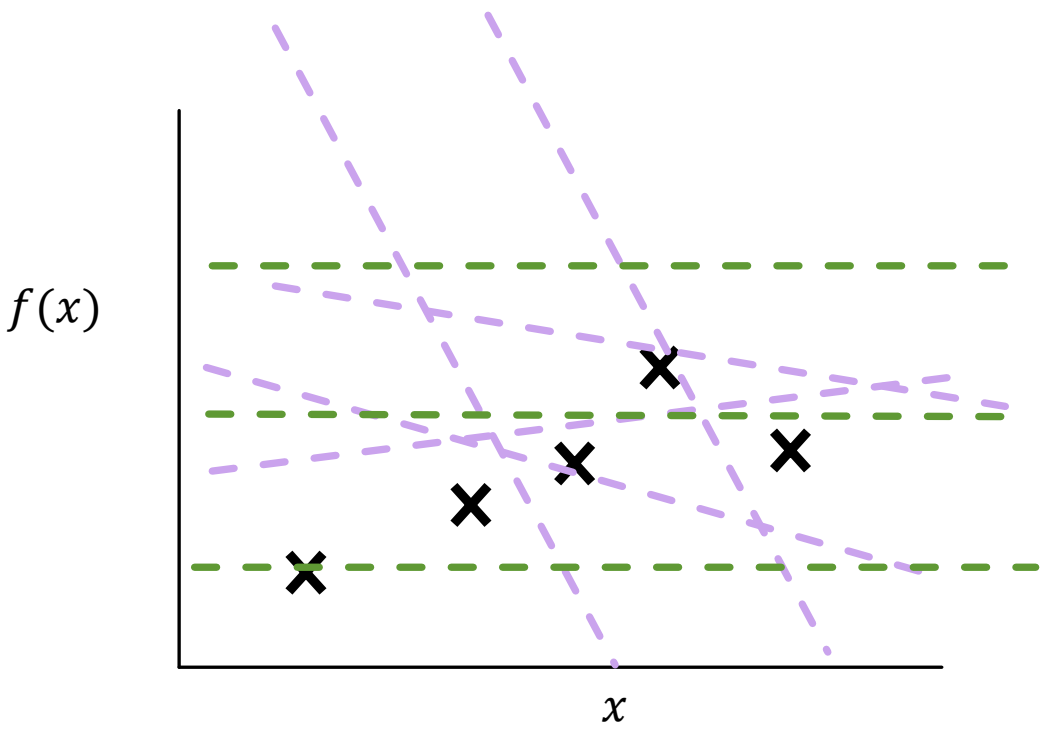
- Restriction bias (aka Absolute bias)
  - Representational power of algorithm
  - The set of hypothesis that an algorithm will consider
    - E.g., an algorithm may consider a hypotheses of straight lines
    - **But many such straight lines are possible**
- Preference bias (aka Relative bias)
  - Indicates what representation(s) a learning algorithm prefers
  - For example, an algorithm may prefer straight lines parallel to x-axis

Types of Inductive Bias



- Restriction bias (aka Absolute bias)
  - Representational power of algorithm
  - The set of hypothesis that an algorithm will consider
    - E.g., an algorithm may consider a hypotheses of straight lines
    - But many such straight lines are possible
- Preference bias (aka Relative bias)
  - Indicates what representation(s) a learning algorithm prefers
  - For example, an algorithm may prefer straight lines parallel to x-axis

Types of Inductive Bias



- Restriction bias (aka Absolute bias)
  - Representational power of algorithm
  - The set of hypothesis that an algorithm will consider
    - E.g., an algorithm may consider a hypotheses of straight lines
    - But many such straight lines are possible
- Preference bias (aka Relative bias)
  - Indicates what representation(s) a learning algorithm prefers
  - For example, an algorithm may prefer straight lines parallel to x-axis

Inductive Learning

- Inducing a general function from training examples
  - Construct a hypothesis h that best approximates the target function f
  - A hypothesis is consistent if it agrees with all training examples
  - A hypothesis is said to generalize well if correctly predicts the label for new examples
  - Learning can be seen as refining the hypothesis space
- Inductive learning is an ill-posed problem
  - **Unless we see all possible examples (data samples) related to the problem statement, inductive learning algorithm can't find an unique hypothesis that for all new examples**

Inductive Learning

- Inductive learning is an ill-posed problem
  - Unless we see all possible examples (data samples) related to the problem statement, inductive learning algorithm can't find an unique hypothesis that for all new examples
  - **But, we can't see all the examples**
  - **So, we have to make an assumption**

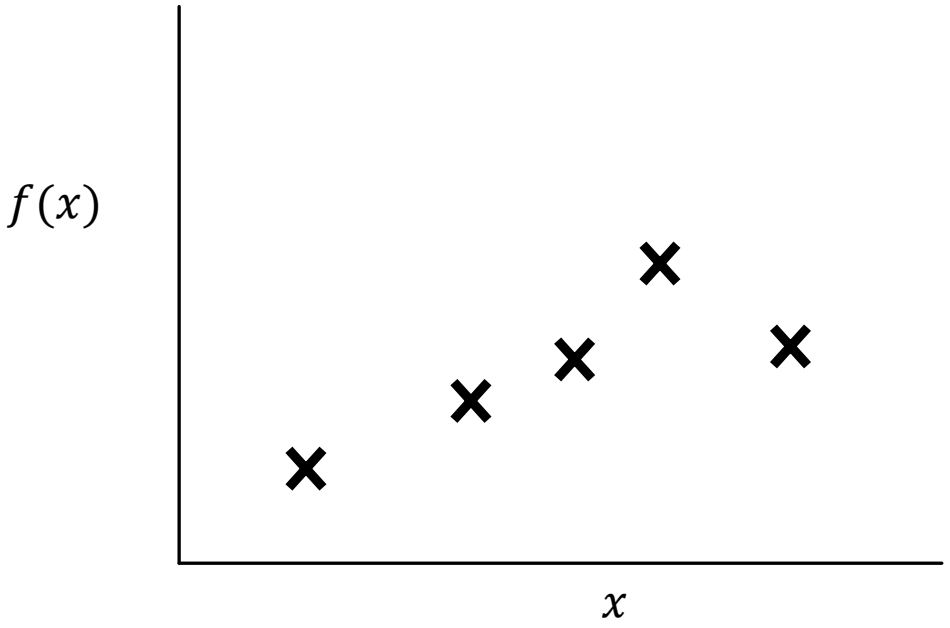
Assumption in Inductive Learning

A hypothesis h that approximates the target function f well in a sufficiently large set of training examples, will also approximate the target function well over other unobserved (new) examples.

Tasks in ML

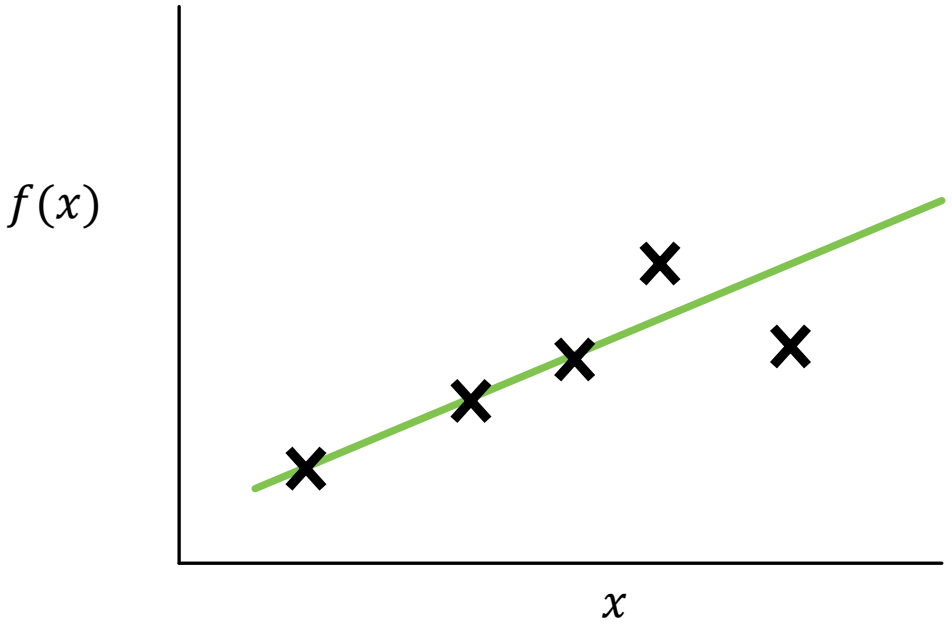
- **Come up with a good hypothesis space**
- **Find an algorithm that works well in that hypothesis space**
- **Generalizability in the future data points**
- **Analyze the confidence in the results (statistical analysis)**
- **Analyze computational tractability**

Some Training Data

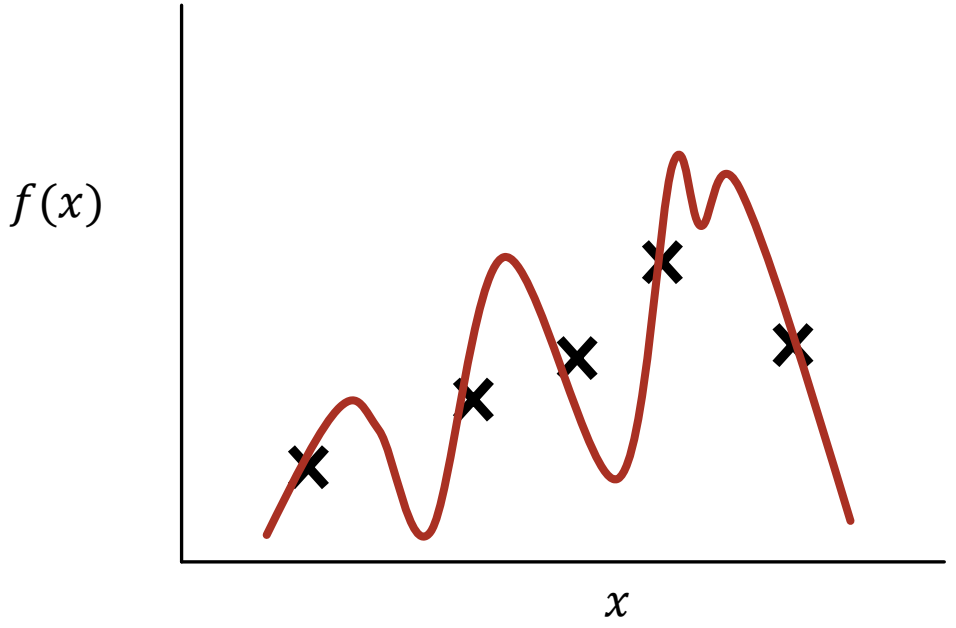


First Set of Training Data

Fitting Possible Curves on the Training Data

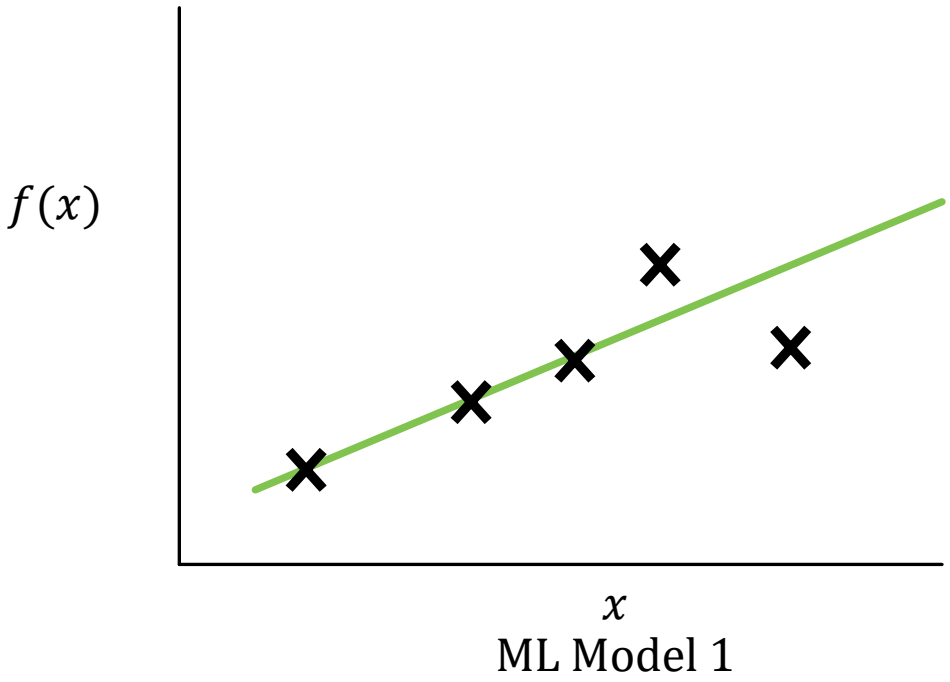


ML Model 1

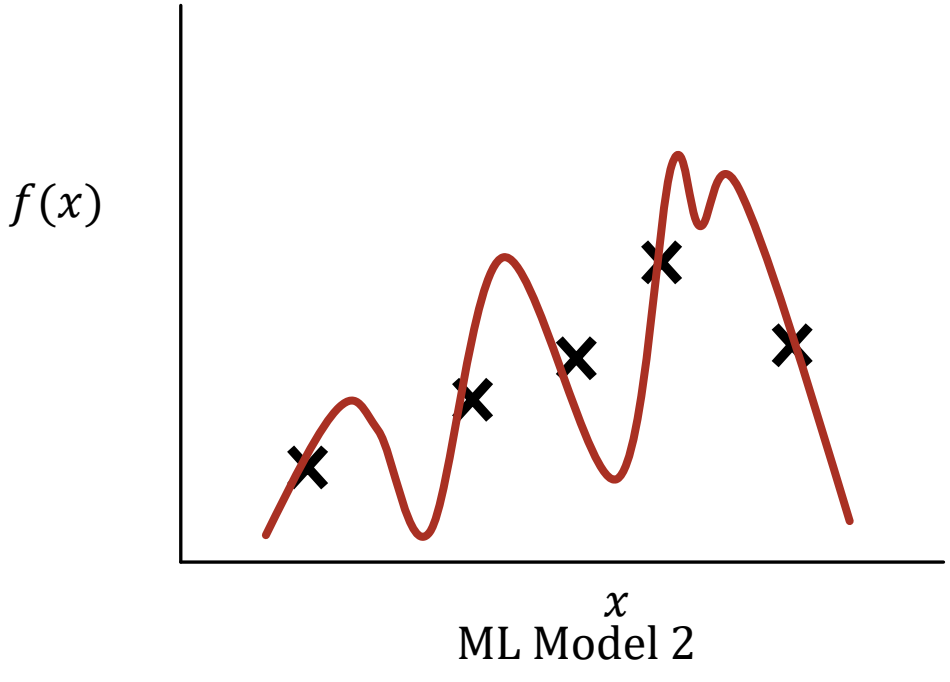


ML Model 2

Fitting Possible Curves on the Training Data



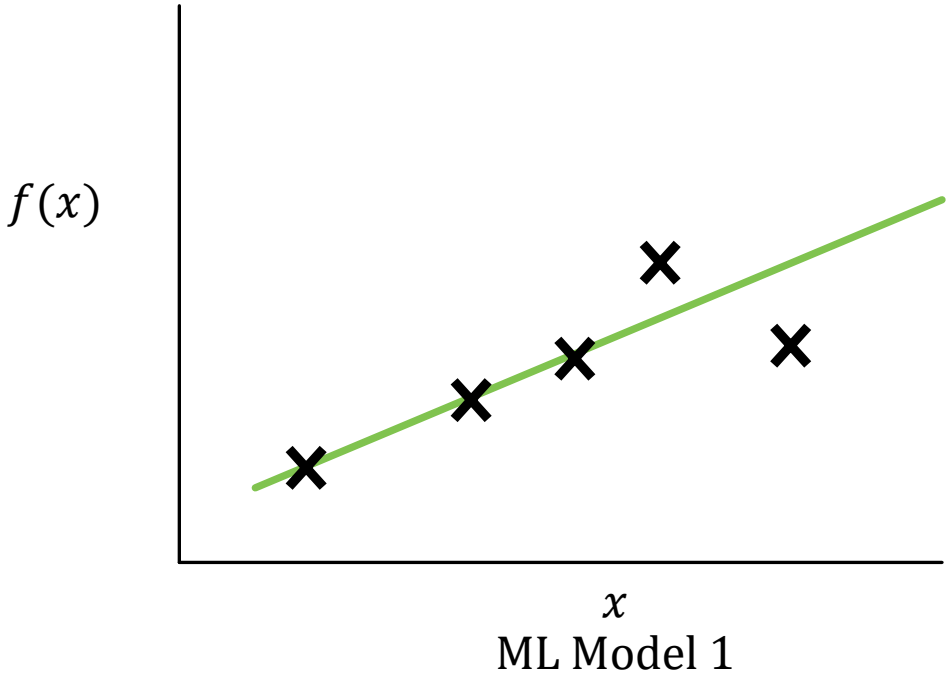
ML Model 1



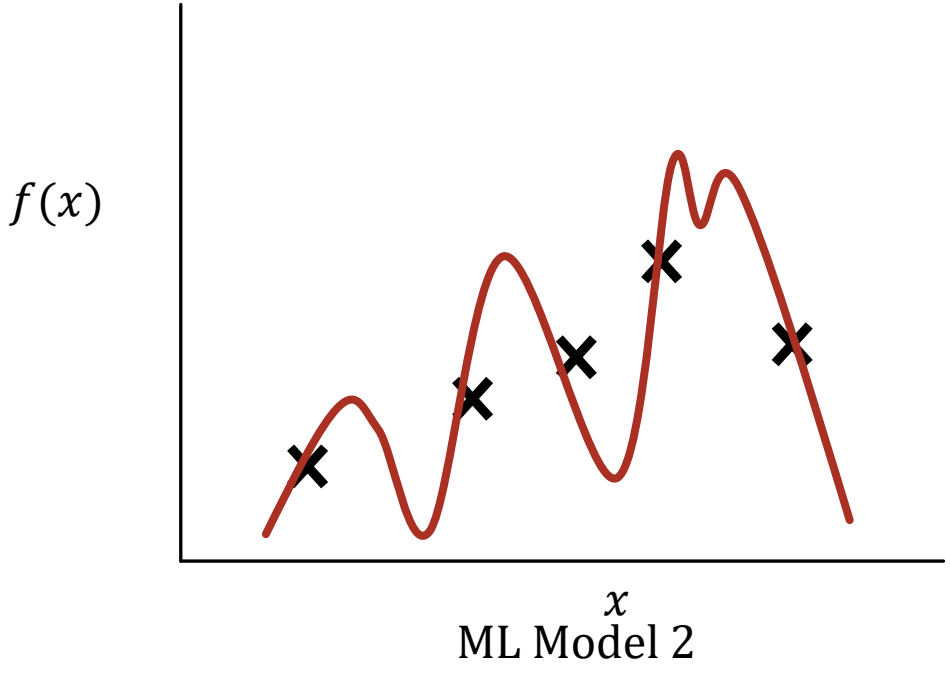
ML Model 2

- More assumption: simple model/ simple curve
  - More bias
- Ranking of bias (high to low)
  - Green curve (model 1)
  - Red curve (model 2)

Fitting Possible Curves on the Training Data



ML Model 1

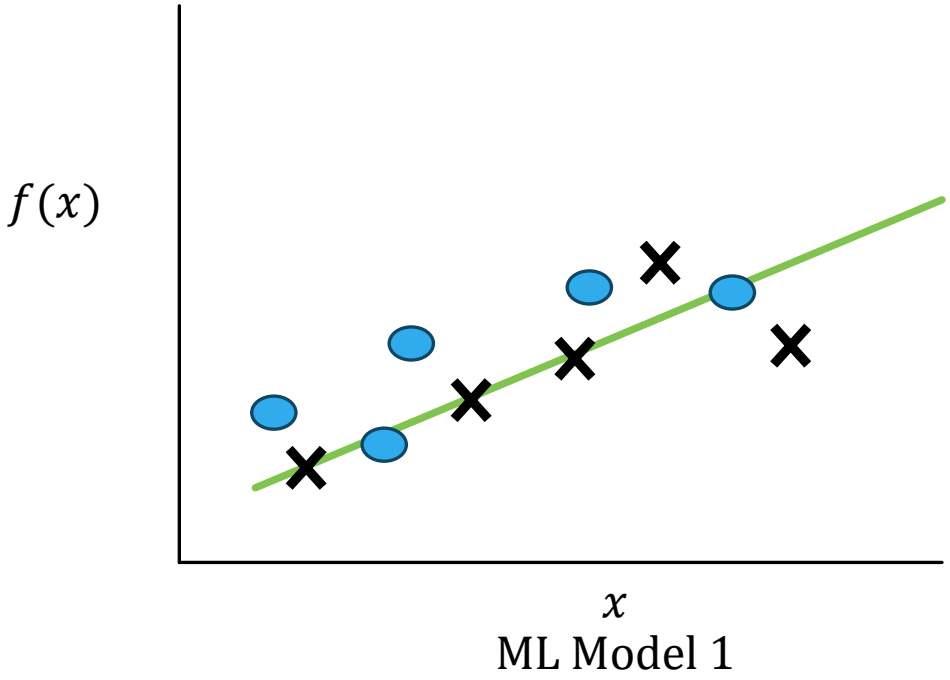


ML Model 2

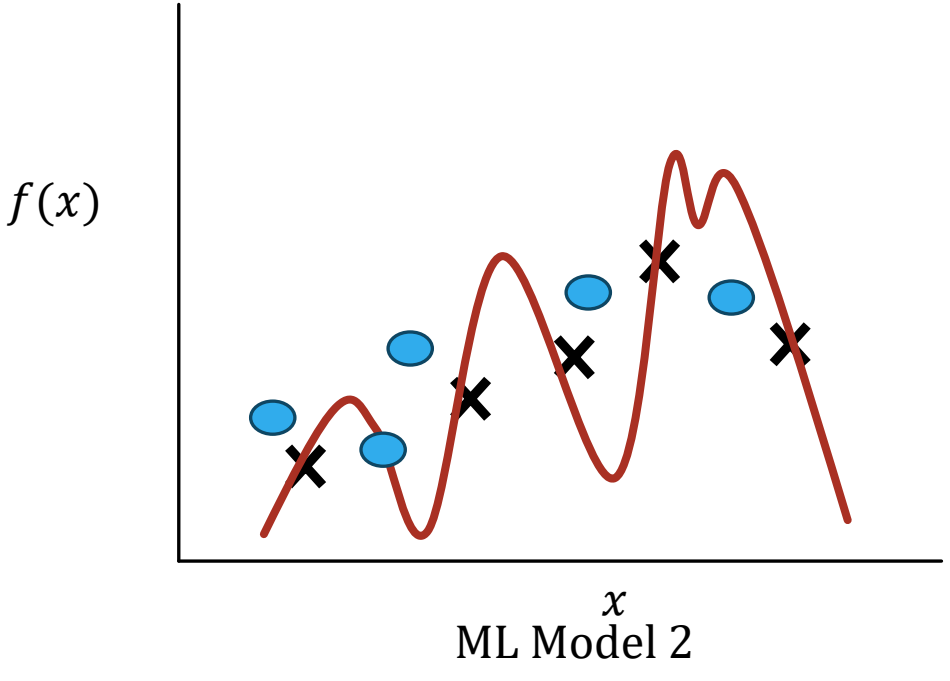
- Ranking of bias (high to low)
  - Green curve (model 1)
  - Red curve (model 2)

- Which curve has more error when compared to training examples?
- Ranking of training error (high to low)
  - Green curve
  - Red curve

Now, Let's See How They Perform on Test Data 1



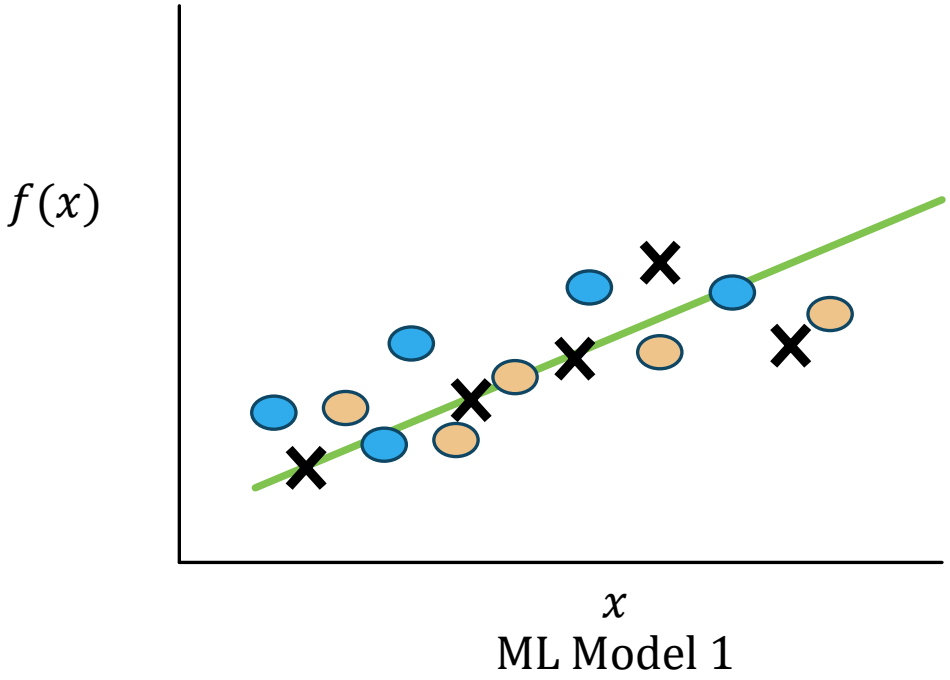
ML Model 1



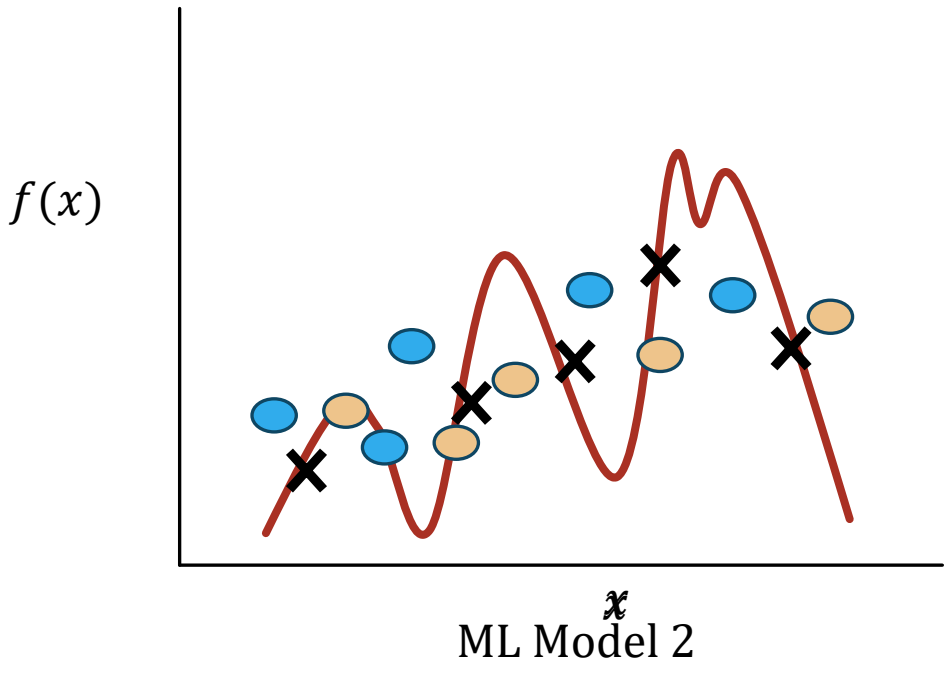
ML Model 2

- Model 1
  - Similar to its performance on training data
  - Test error $e_g(1)$
- Model 2
  - Very different from its performance on training data
  - Test error $e_r(1)$

Now, Let's See How They Perform on Test Data 2



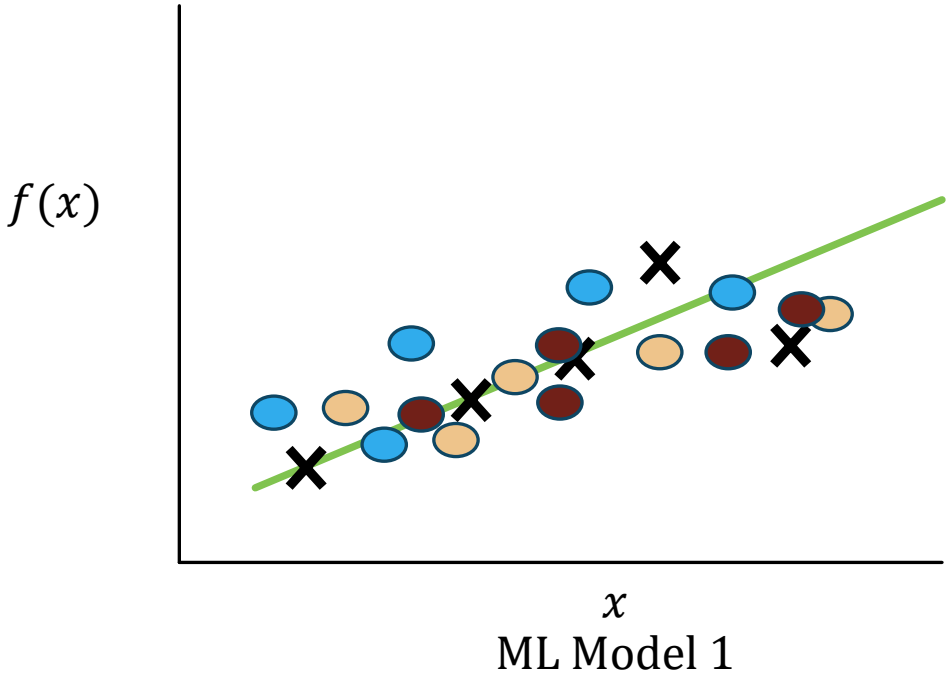
ML Model 1



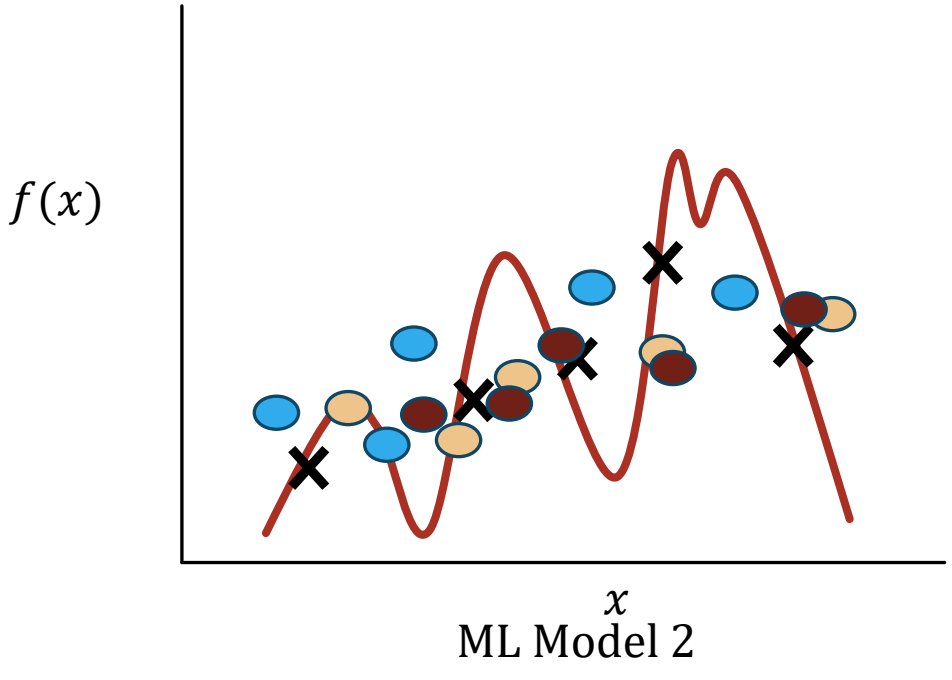
ML Model 2

- Model 1
  - Similar to its performance on training data, and test data 1
  - Test error $e_g(2)$
- Model 2
  - Very different from its performance on training data, and test data 1
  - Test error $e_r(2)$

Now, Let's See How They Perform on Test Data 3



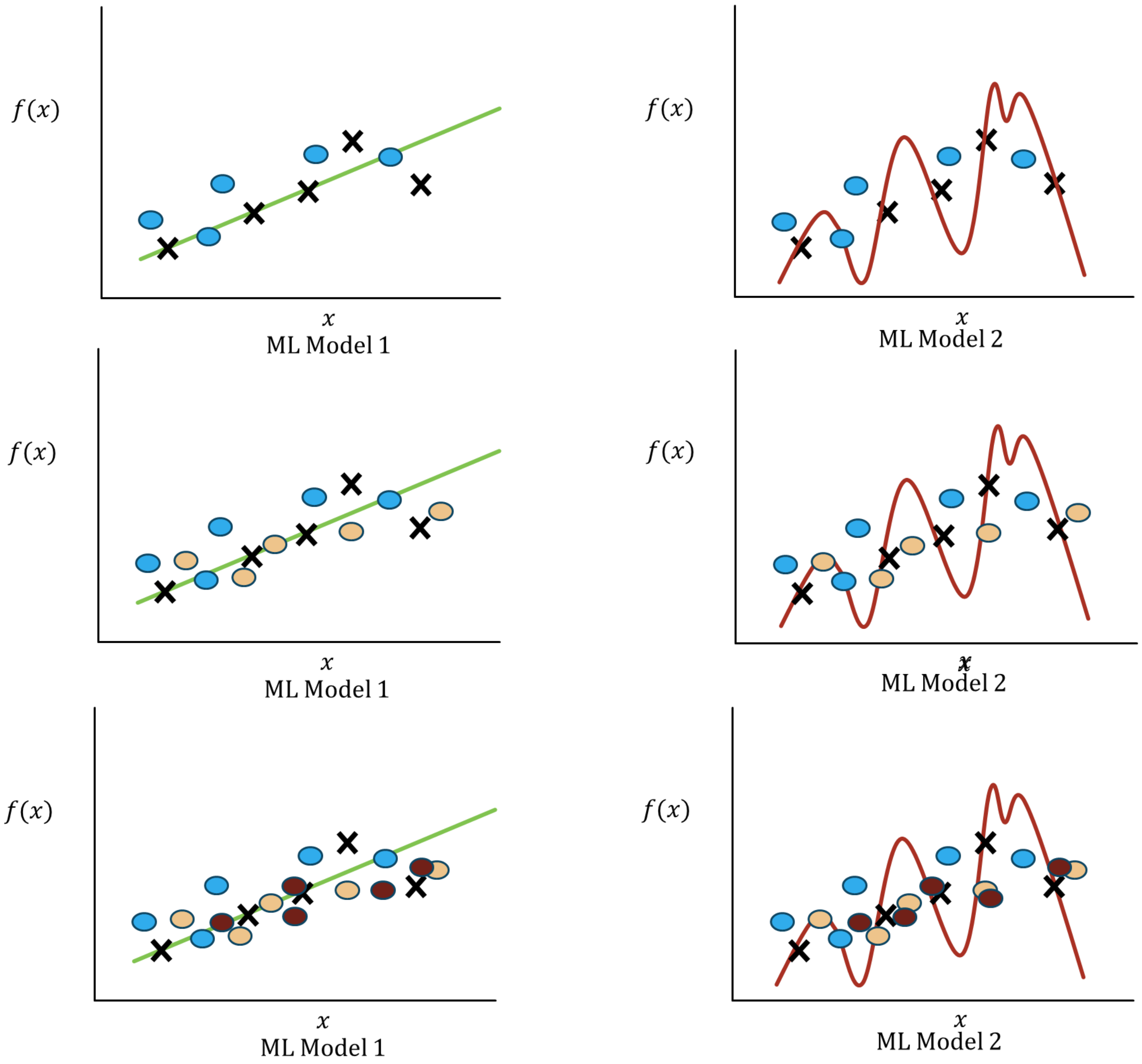
ML Model 1

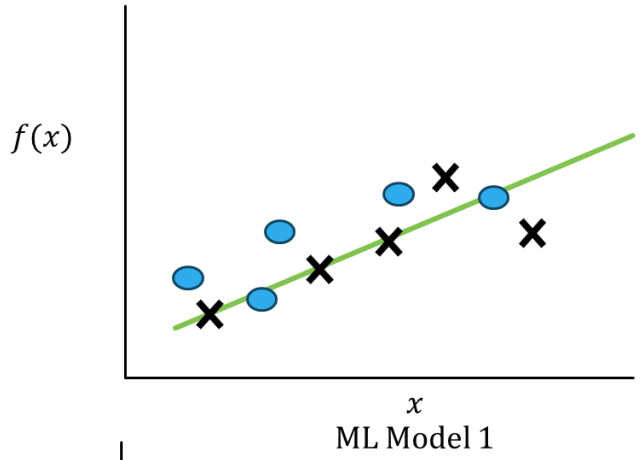


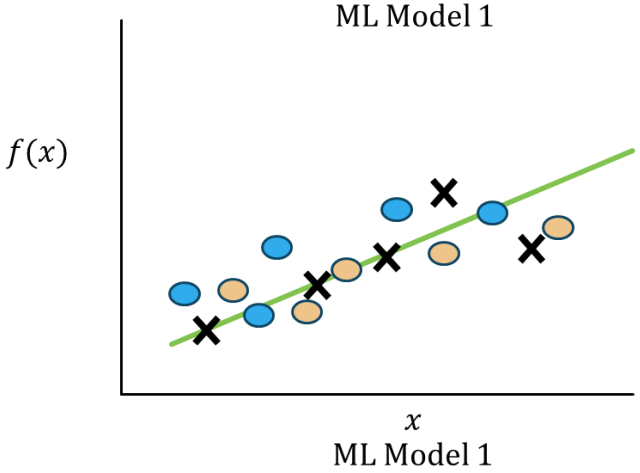
ML Model 2

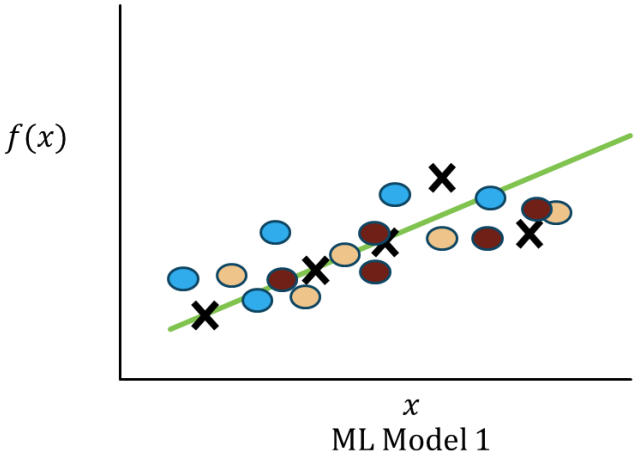
- Model 1
  - Similar to its performance on training data, test data 1, and test data 2
  - Test error $e_g(3)$
- Model 2
  - Very different from its performance on training data, test data 1, and test data 2
  - Test error $e_r(3)$

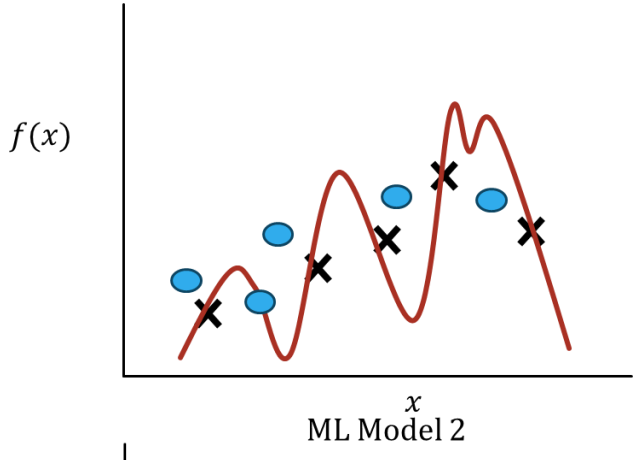
So, What do We Observe?

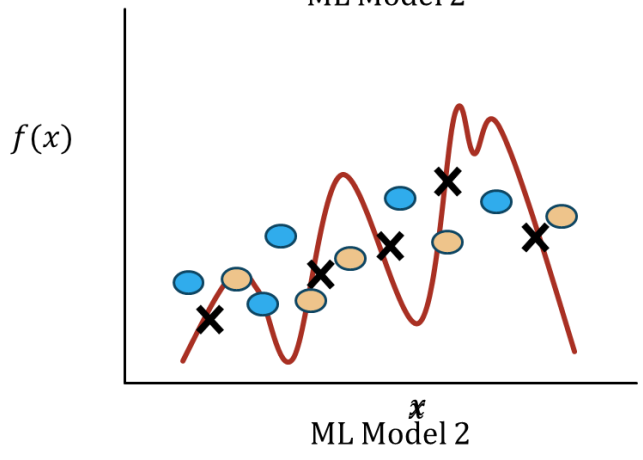


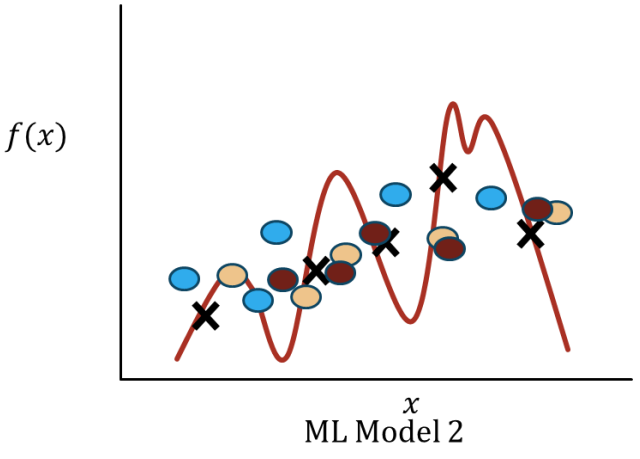








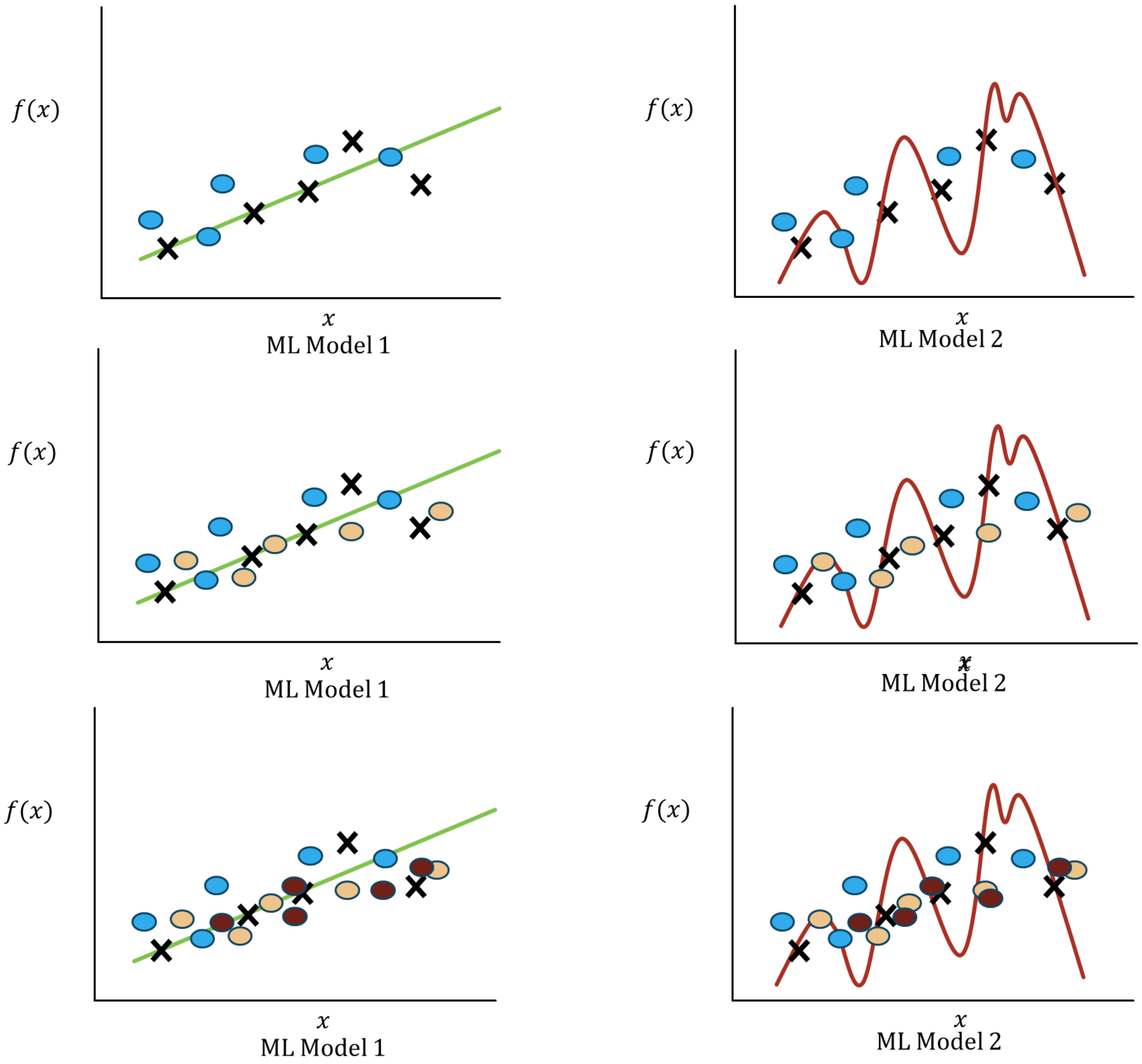




So, What do We Observe?

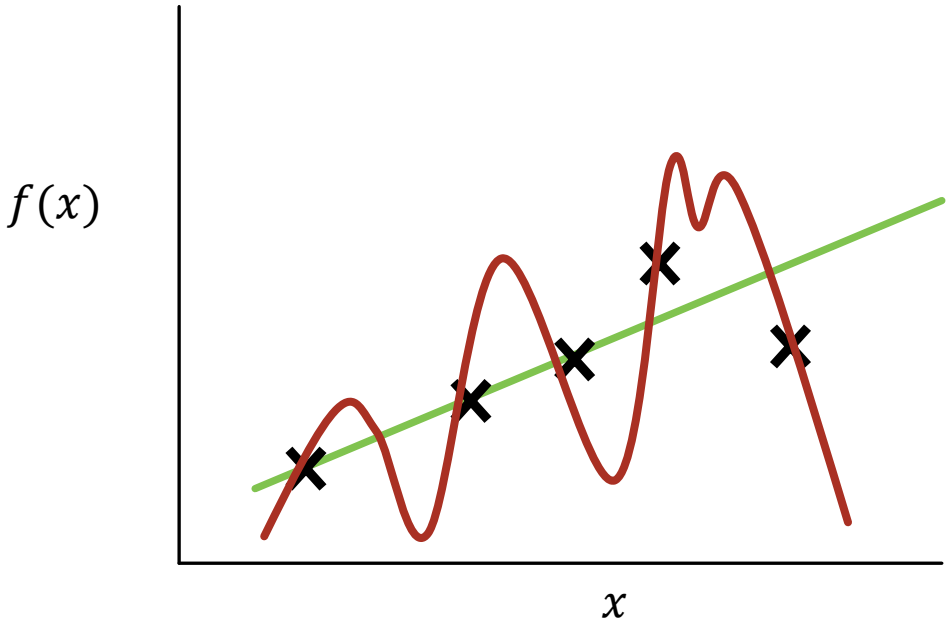
- The three sets of test data are similar to the training data with slight differences
- For the green curve (model 1), the errors in the test data $e_g(1), e_g(2), e_g(3)$ are similar to each other and similar to training error
  - Variance of the error is low
  - Low variance
  - High bias

So, What do We Observe?



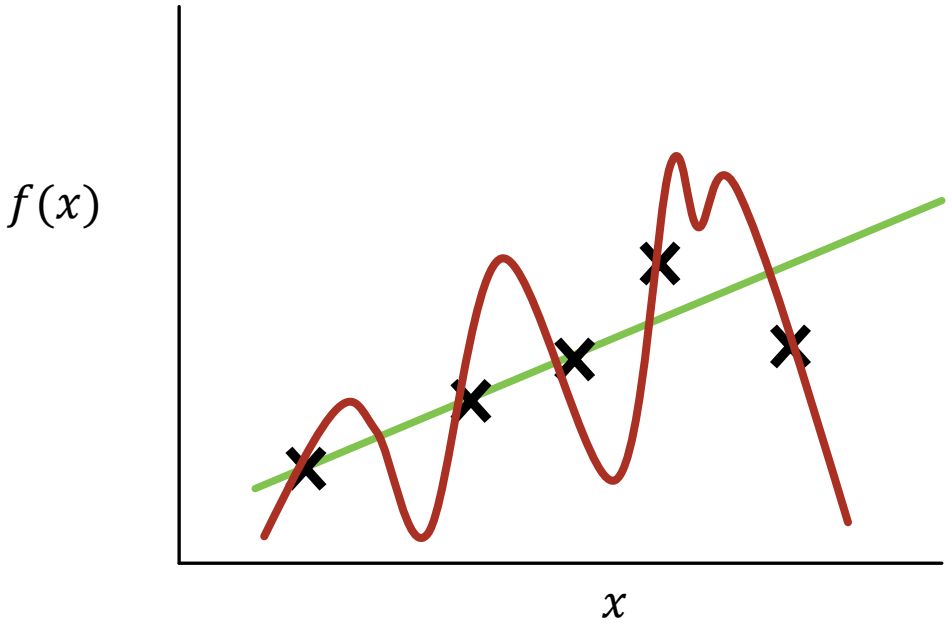
- The three sets of test data are similar to the training data with slight differences
- For the red curve (model 2), the errors in the test data $e_r(1), e_r(2), e_r(3)$ are not similar to each other and not similar to training error
  - Variance of the error is high
  - High variance
  - Low bias

Bias and Variance



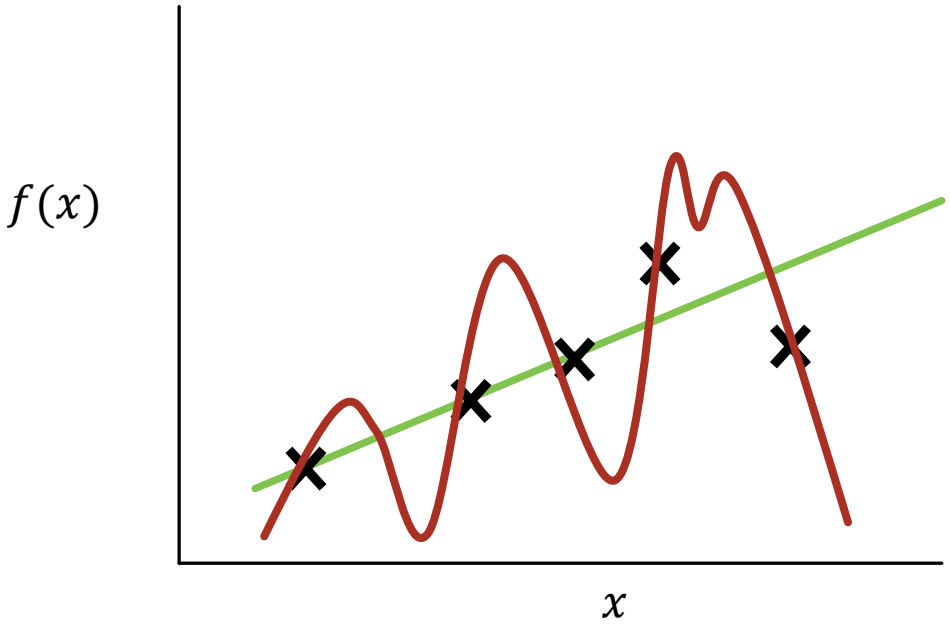
- Bias is the difference between the average prediction of our model and the correct value which we are trying to predict. Model with high bias pays very little attention to the training data and oversimplifies the model. It leads to high error on training and test data.
- Variance is the variability of model prediction across different sets of data

Bias and Variance



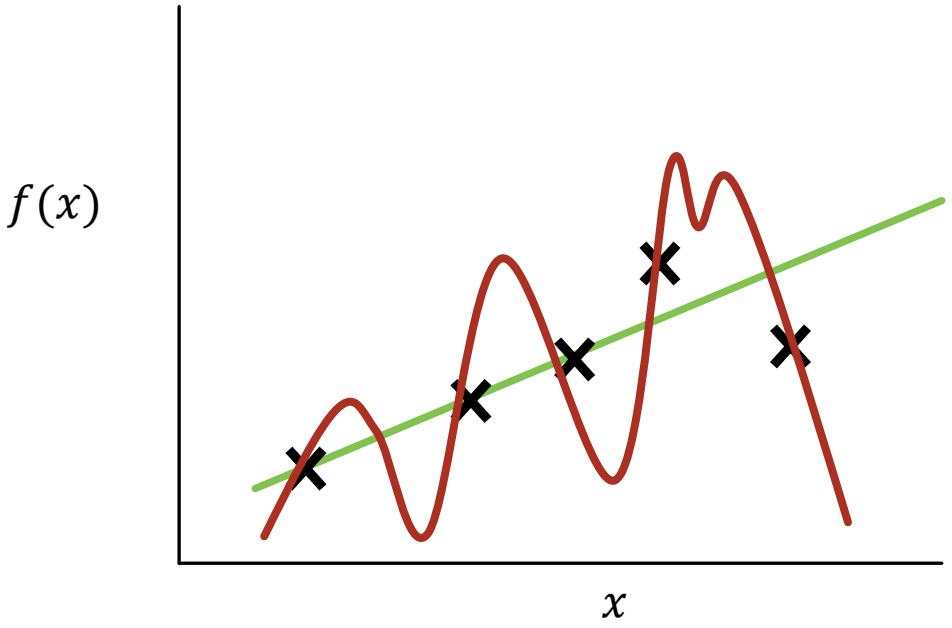
- More assumption: simple model/ simple curve
  - More bias, less variance
- Less assumption: complex model/ simple curve
  - Less bias, more variance

Impact of Bias and Variance



- More assumption: simple model/ simple curve
  - More bias, less variance
  - Underfitting
  - Consistent but relative poorer performance across datasets
- Less assumption: complex model/ simple curve
  - Less bias, more variance
  - Overfitting
  - Inconsistent performance across datasets

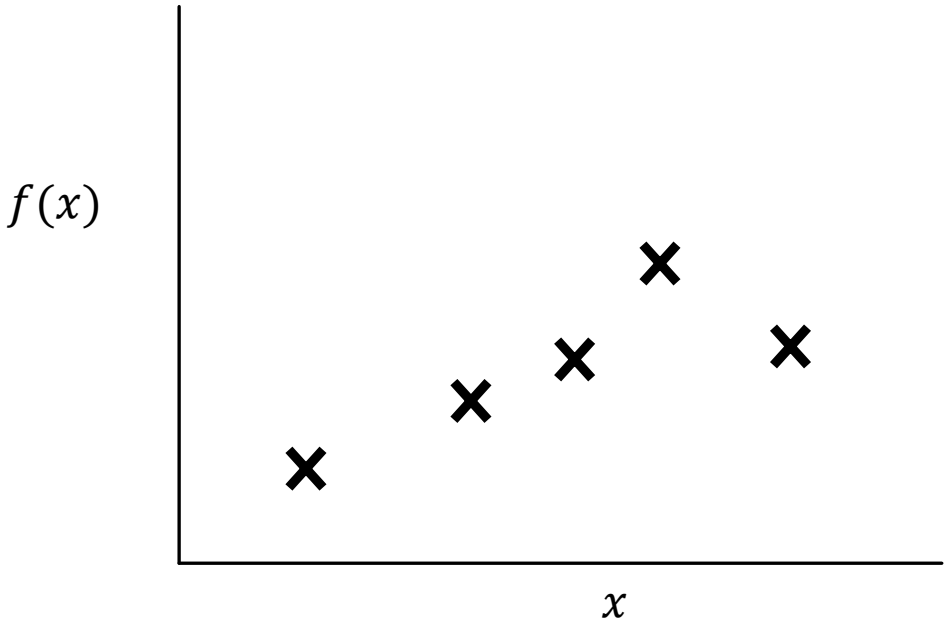
Impact of Bias and Variance



- More assumption: simple model/ simple curve
  - More bias, less variance
  - Underfitting
  - Consistent but relative poorer performance across datasets
- Less assumption: complex model/ simple curve
  - Less bias, more variance
  - Overfitting
  - Inconsistent performance across datasets

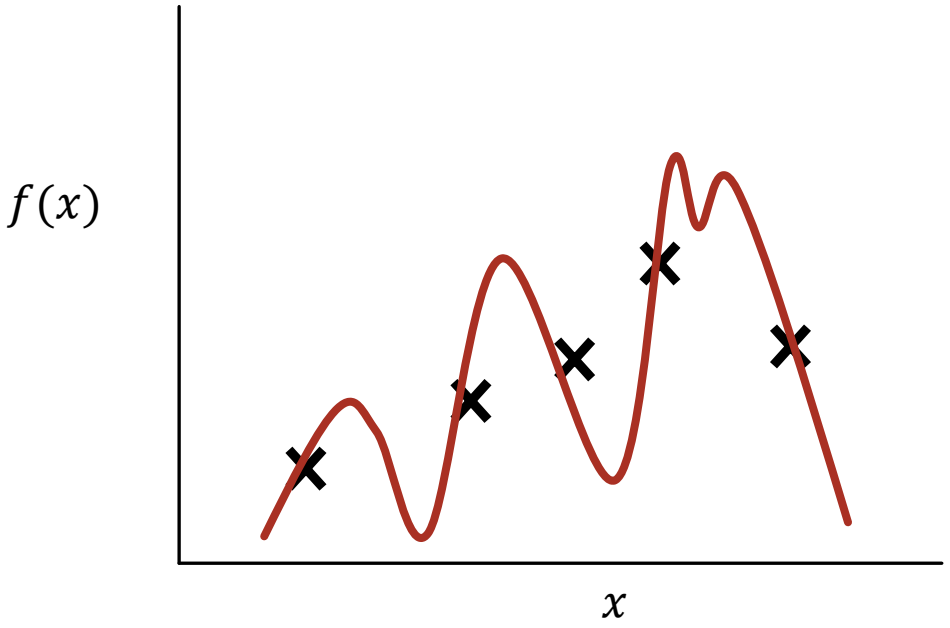
So, we want low bias and low variance in any model. But we can't achieve these at the same time

Another Aspect of Overfitting



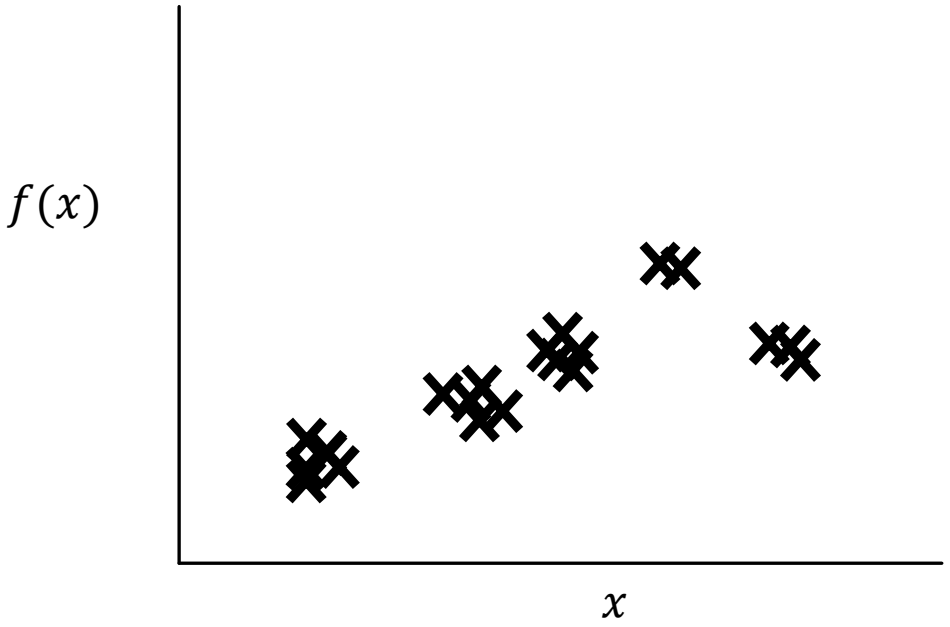
- Suppose, you have a small amount of training data
- Can you design a model to fit all the training data?

Another Aspect of Overfitting



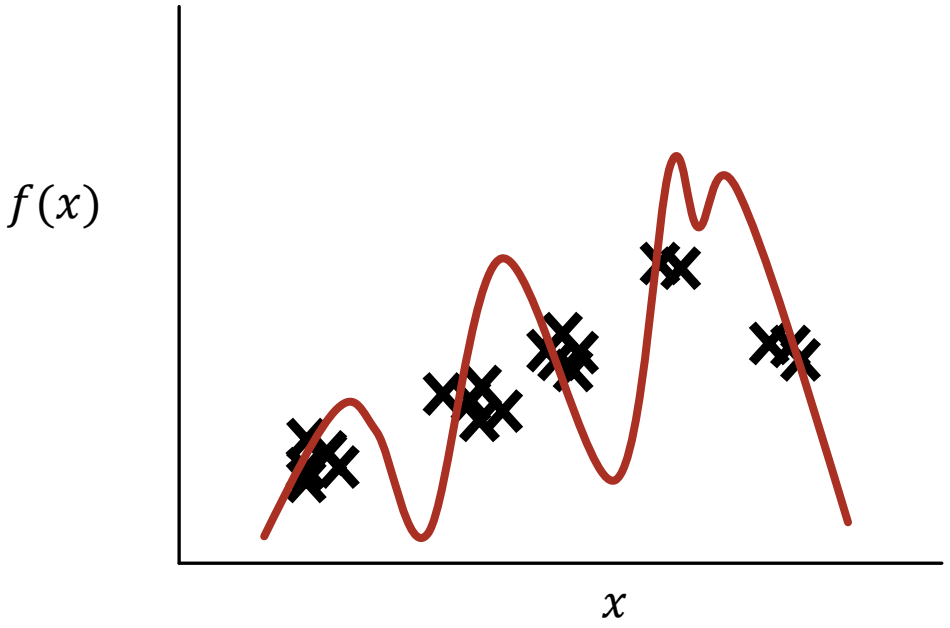
- Suppose, you have a small amount of training data
- Can you design a model to fit all the training data?
  - Yes (more or less easily)

Another Aspect of Overfitting



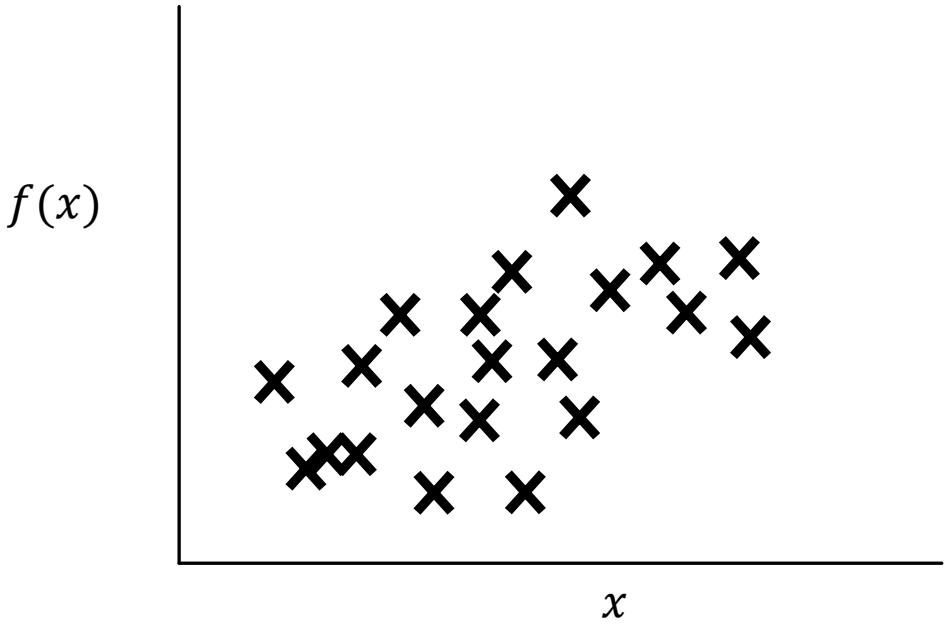
- Now, suppose, you have a lot of training data but of similar types
- Can you design a model to fit all the training data?

Another Aspect of Overfitting



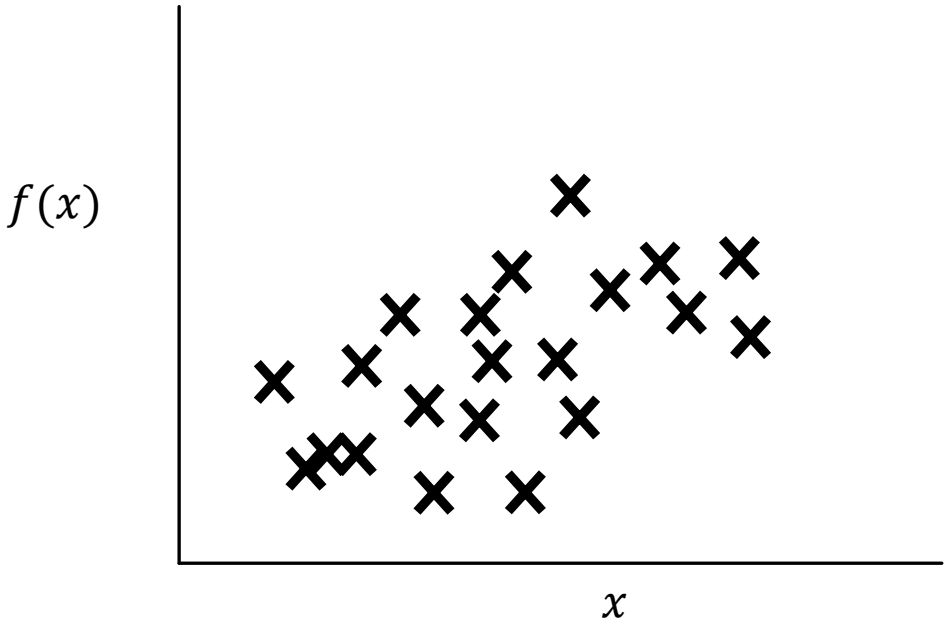
- Now, suppose, you have a lot of training data but of similar types
- Can you design a model to fit all the training data?
  - Yes (more or less)

Another Aspect of Overfitting



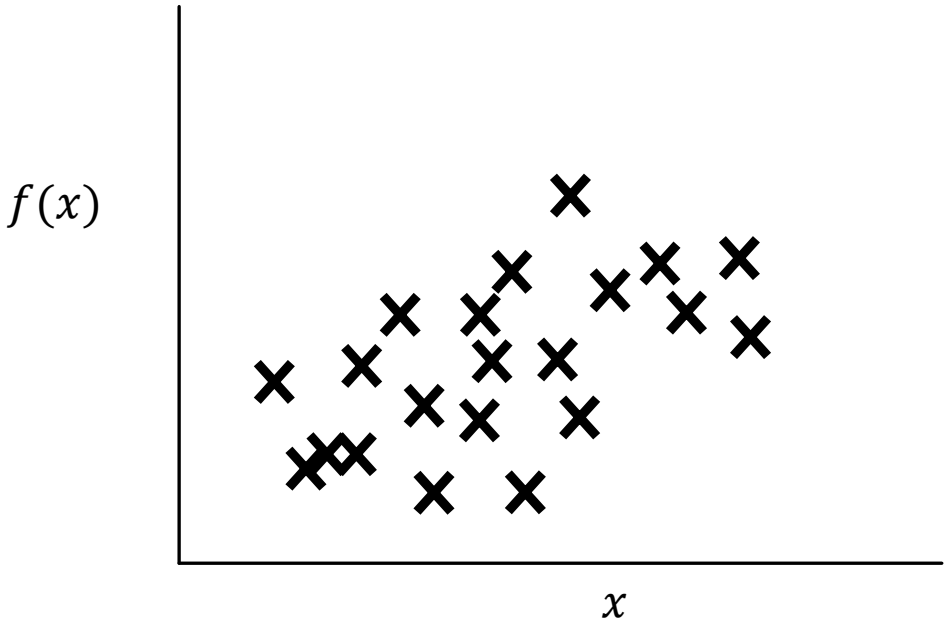
- Now, suppose, you have a lot of training data of sufficient diversity
- Can you design a model to fit all the training data?

Another Aspect of Overfitting



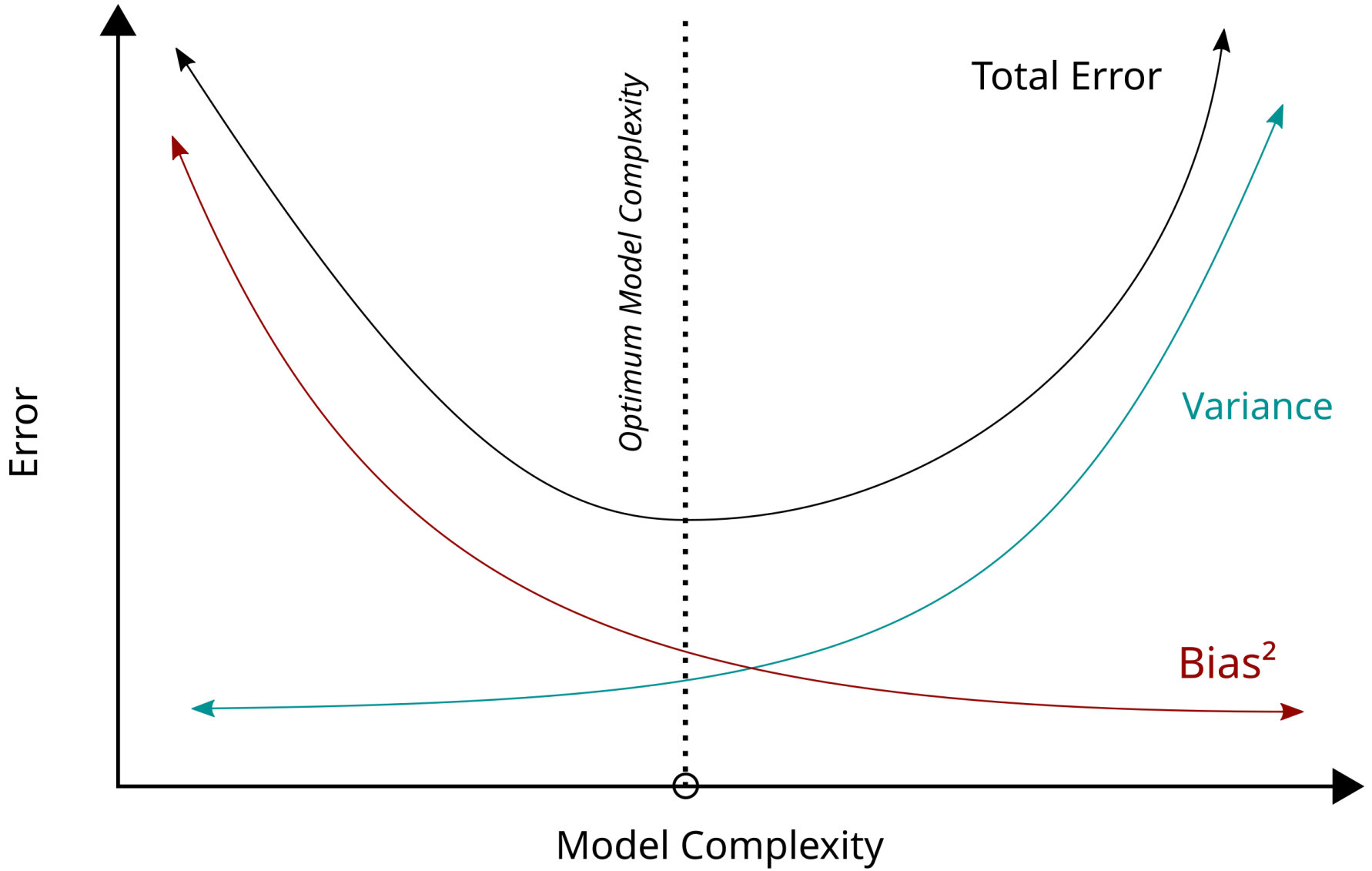
- Now, suppose, you have a lot of training data of sufficient diversity
- Can you design a model to fit all the training data?
  - Very difficult and practically almost impossible

Another Aspect of Overfitting



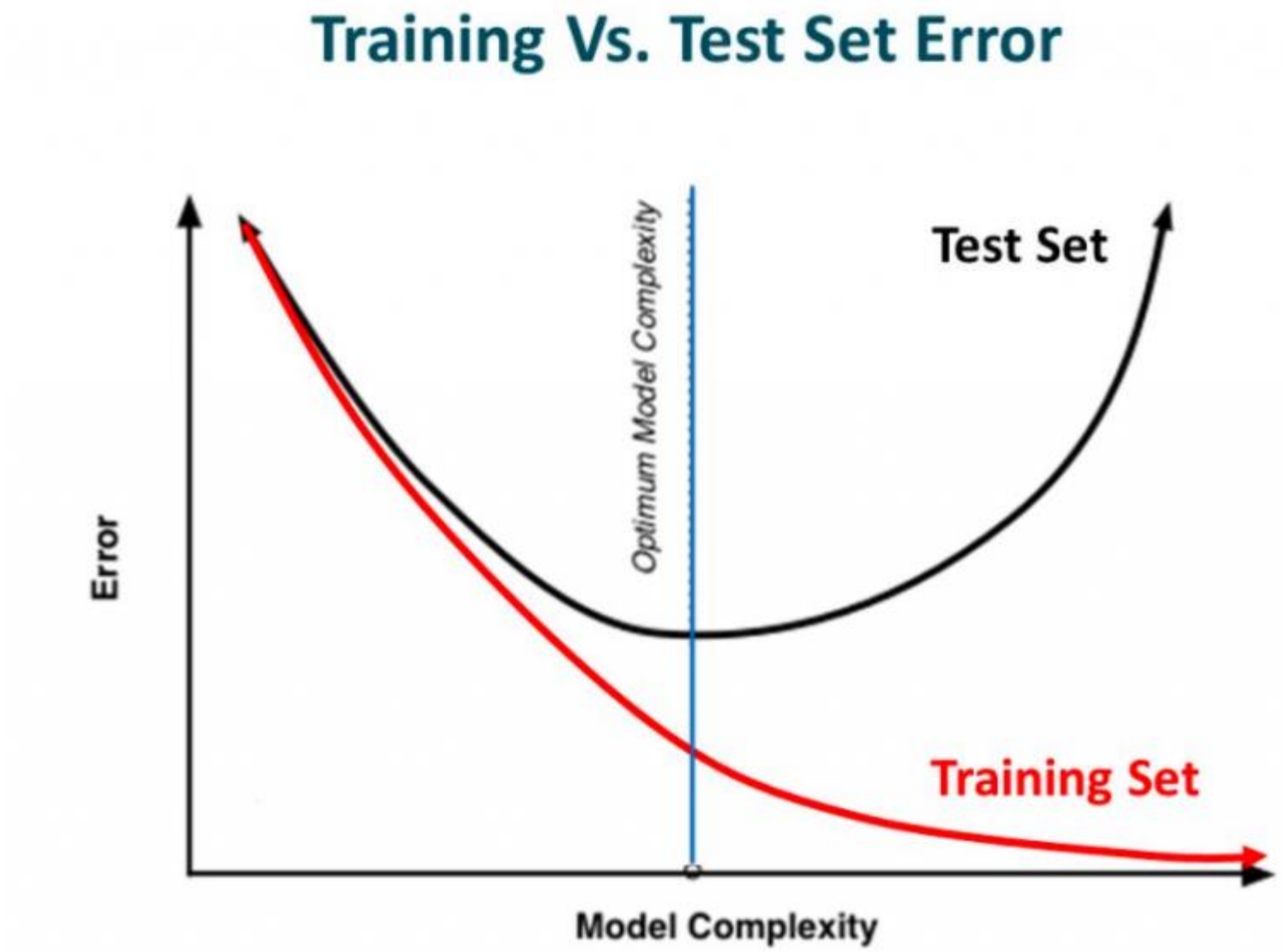
- Training data of sufficient diversity
  - Prevents overfitting

Errors



https://en.wikipedia.org/wiki/Bias%E2%80%93variance_tradeoff#/media/File:Bias_and_variance_contributing_to_total_error.svg

Errors

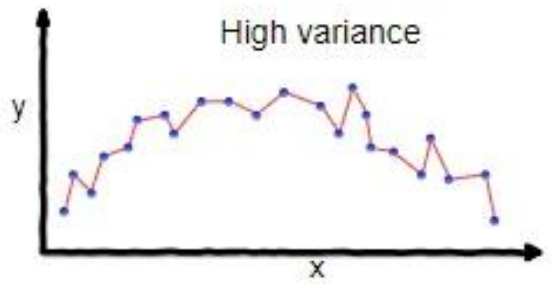


https://www.kaggle.com/code/sid321axn/regularization-techniques-in-deep-learning/notebook

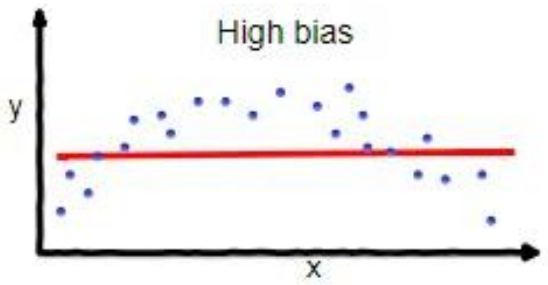
Bias Variance Tradeoff

- We want low bias and low variance

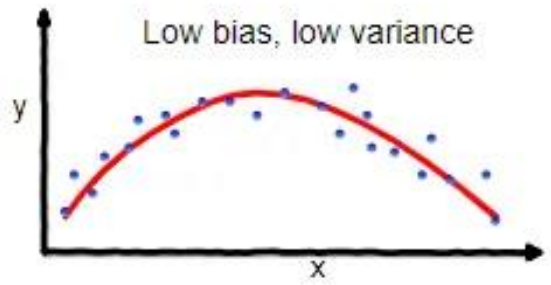
- Need to find a sweet spot between simple and complex model



overfitting



underfitting

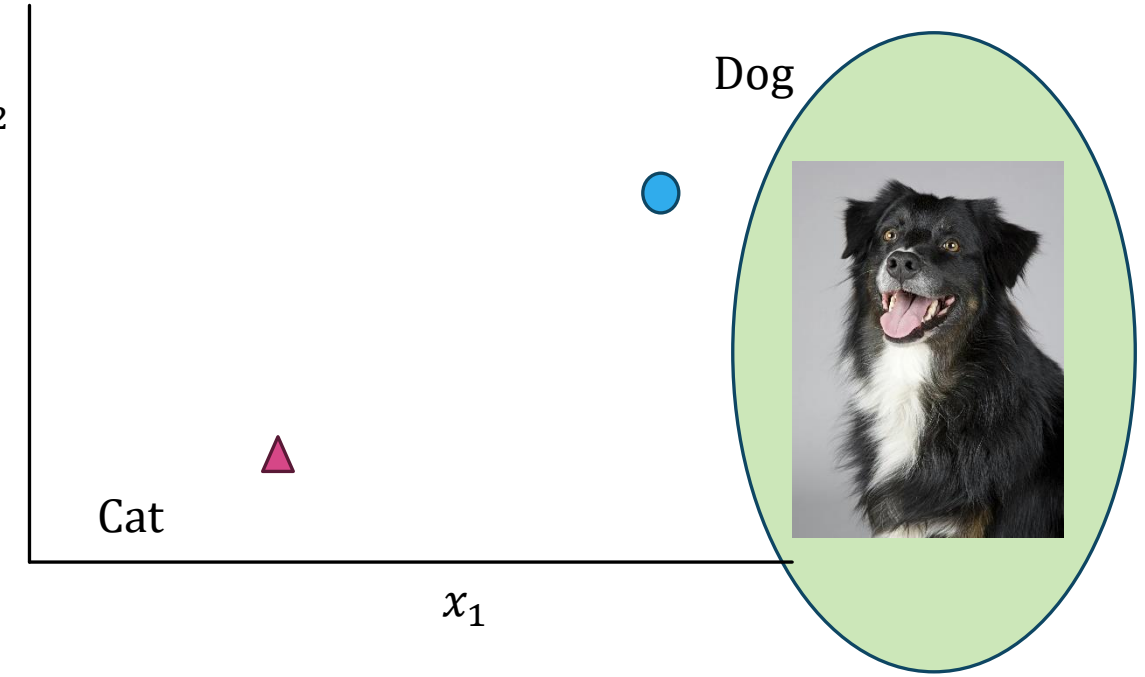


Good balance

https://towardsdatascience.com/understanding-the-bias-variance-tradeoff-165e6942b229

Number of Data points that a Hypothesis Can Classify Correctly

CAT



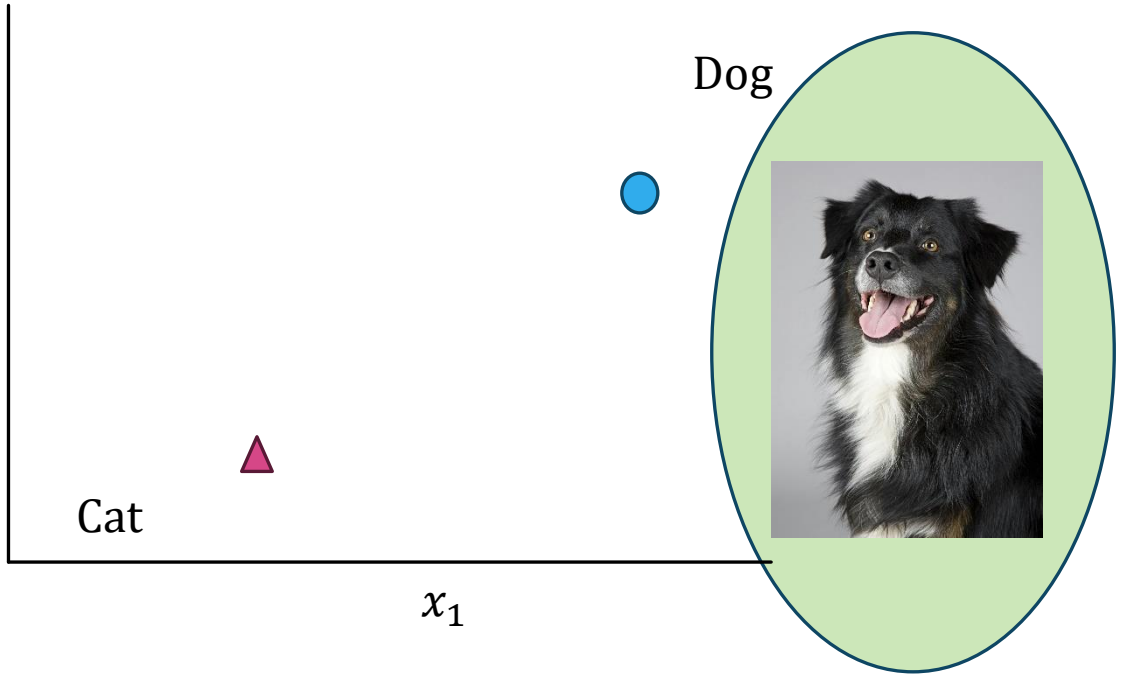
DOG

x_1: Height
x_2: Weight

https://www.pexels.com/

Let's Start With Straight Line

CAT

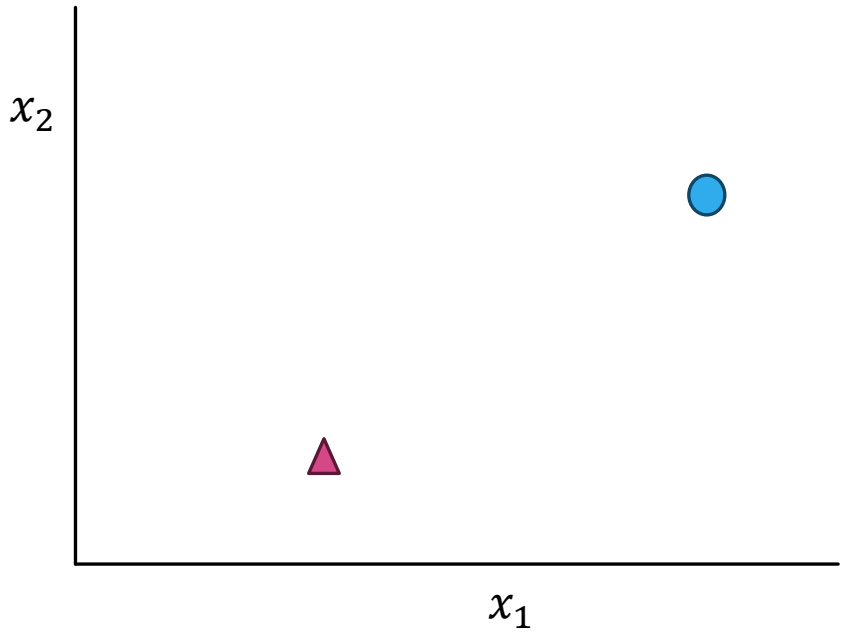


DOG

x_1: Height
x_2: Weight

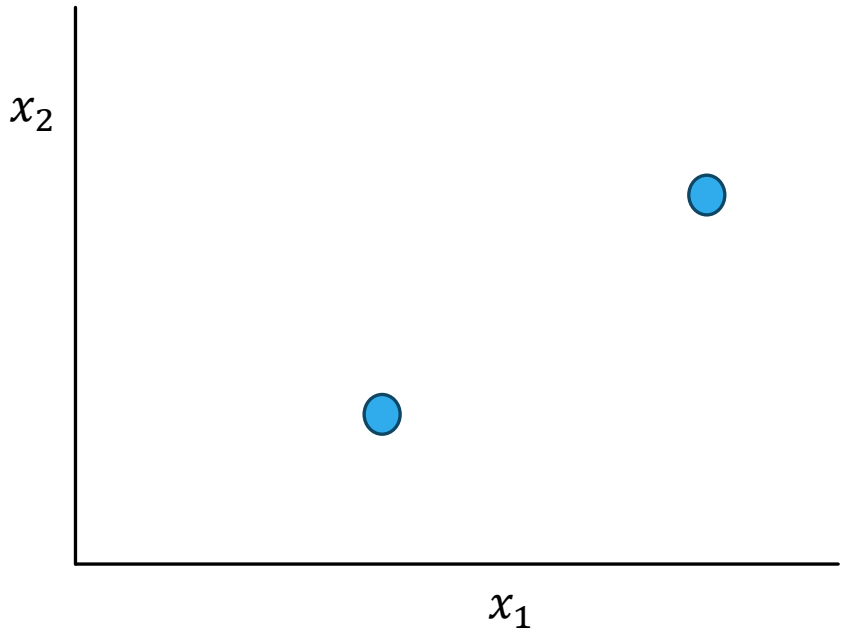
https://www.pexels.com/

Let's Start With Straight Line



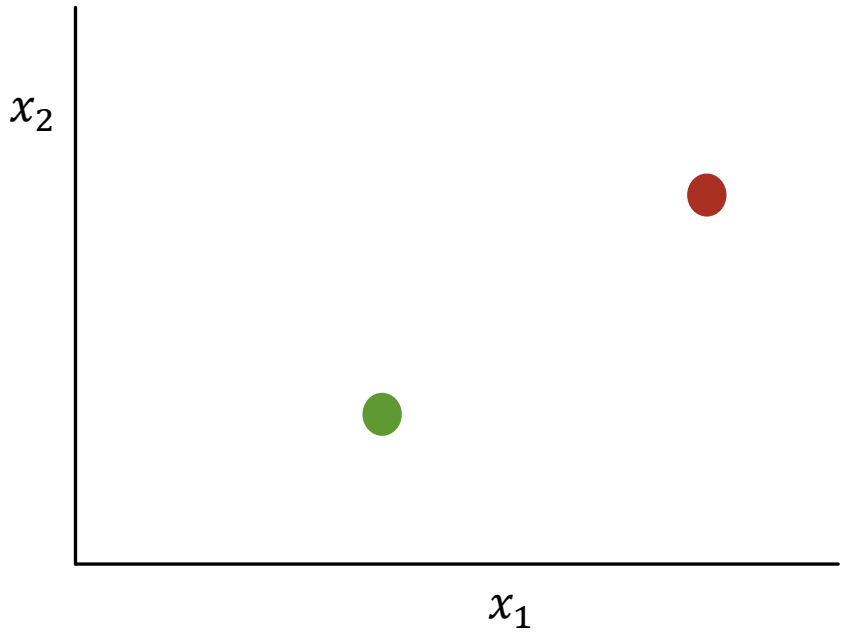
Instead of Cats and Dogs, let's consider positive class and negative class

Let's Start With Straight Line



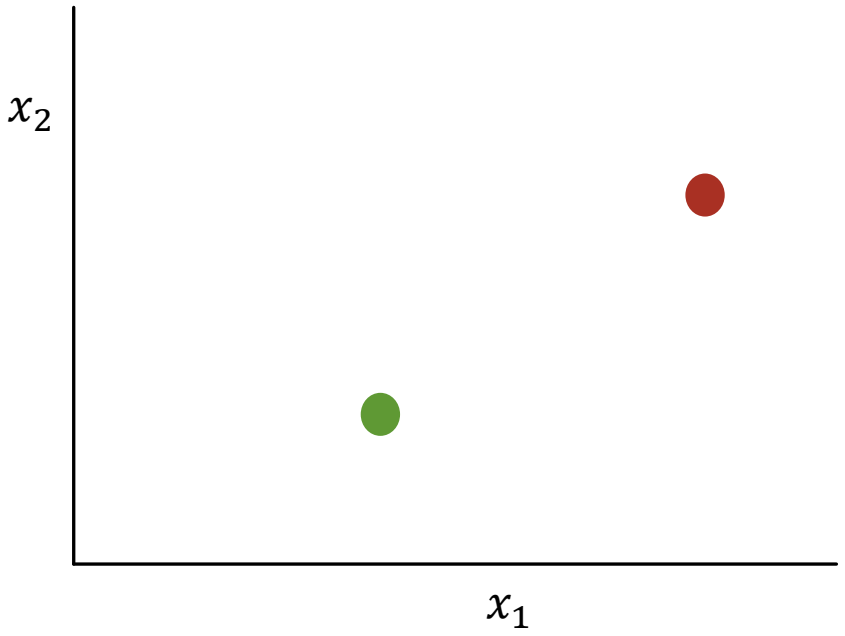
Instead of Cats and Dogs, let's consider positive class and negative class

Let's Start With Straight Line



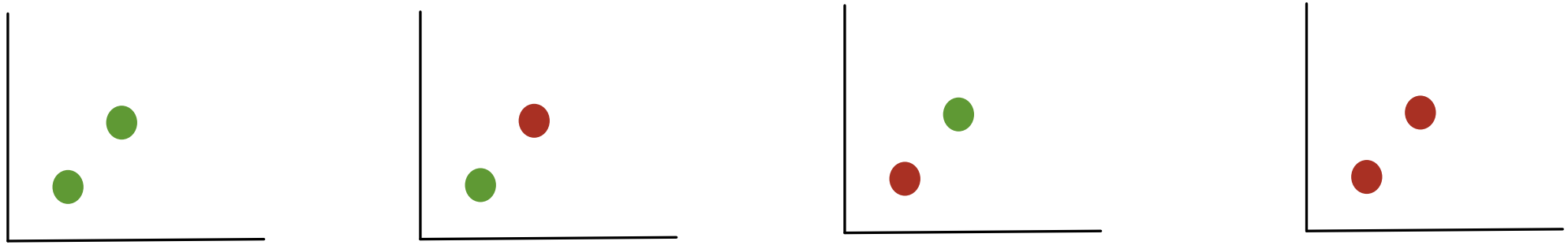
Instead of Cats and Dogs, let's consider positive class (green) and negative class (red)

Let's Start With Straight Line



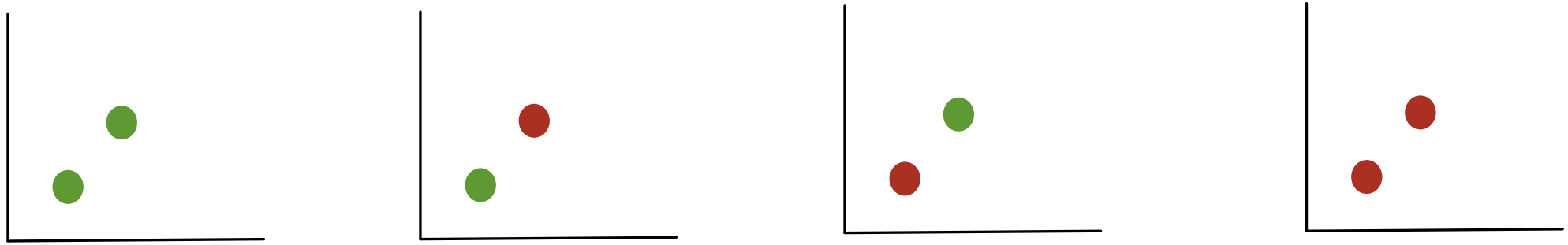
If I have 2 data points, how many combinations of positive and negative class are possible?

Let's Start With Straight Line



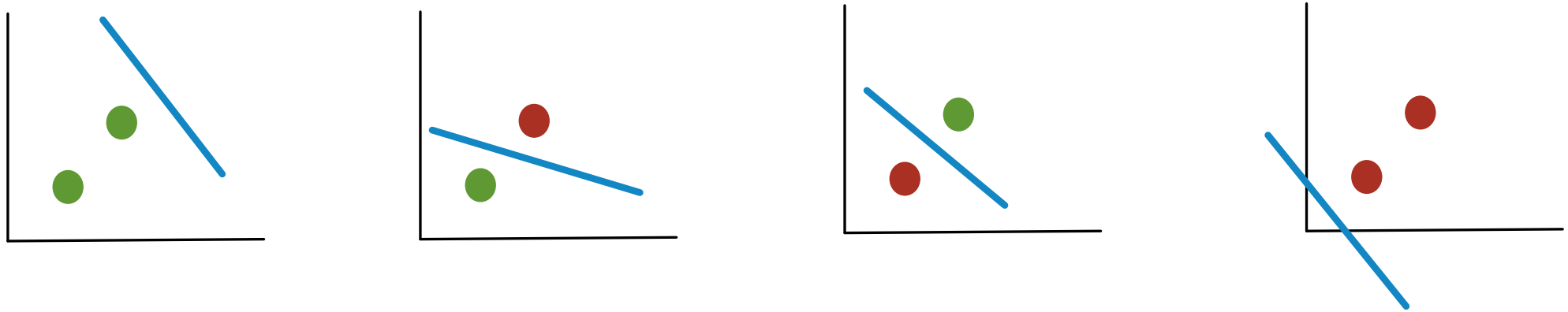
If I have 2 data points, how many combinations of positive and negative class are possible?

Let's Start With Straight Line

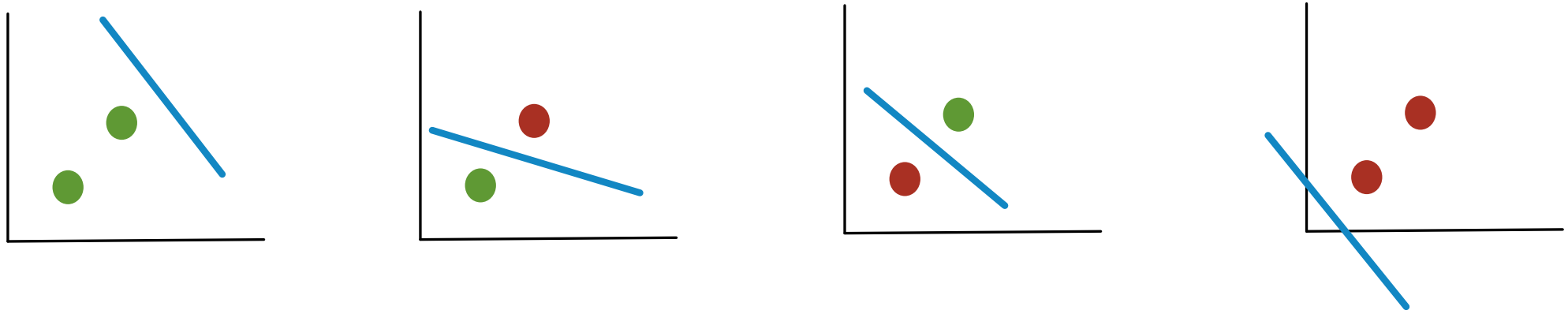


Can I draw straight lines to classify them correctly?

Let’s Start With Straight Line



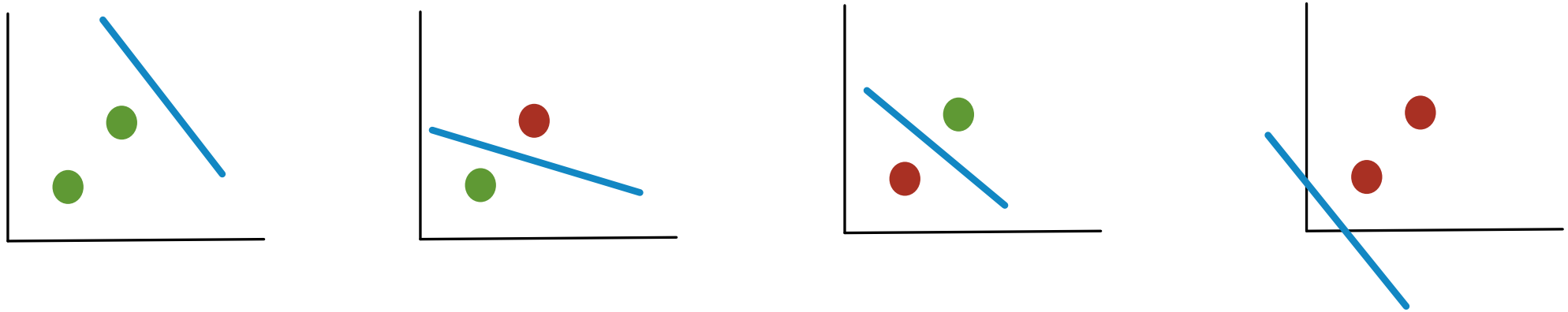
Let's Start With Straight Line



That means the hypothesis space of straight lines is expressive enough to classify two pints in 2D plane

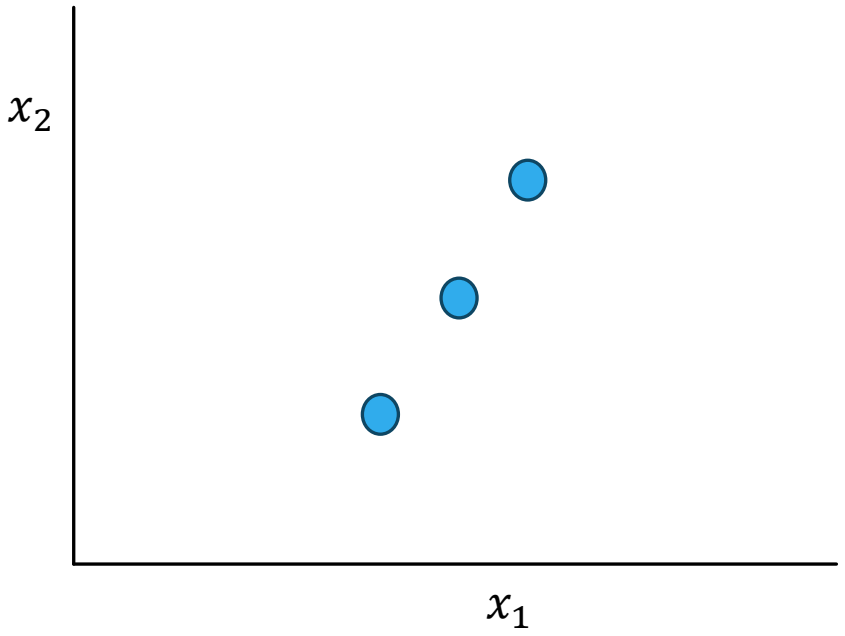
We say that the hypothesis space of straight lines can shatter two points in a 2D plane

Shattering



A set of N points is said to be shattered by a hypothesis space H if there are hypotheses in H that can separate the positive and negative examples in all of the 2^N possible ways for at least one configuration of data points

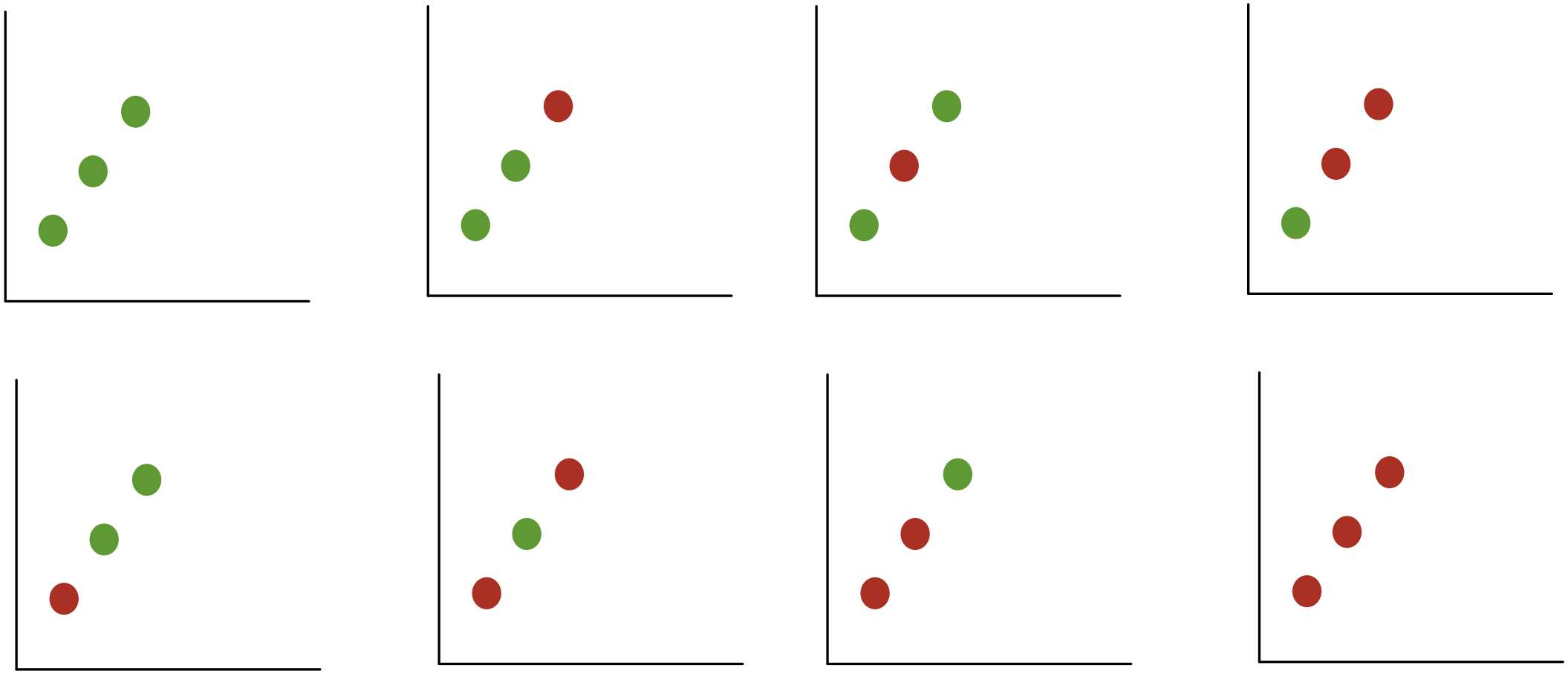
Three Points



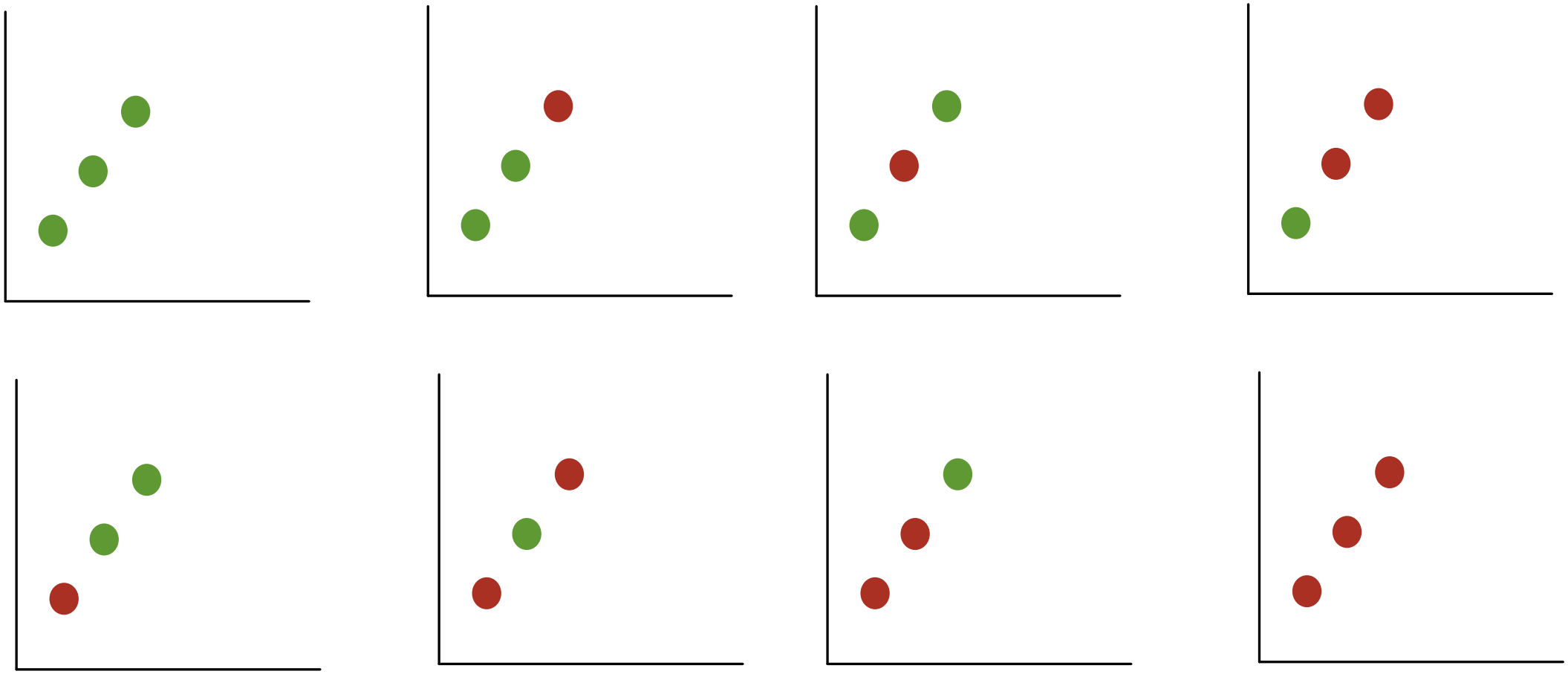
Now consider three points

Can straight lines classify all possible combination of classes for this configuration of 3 points?

Possible Class Combinations

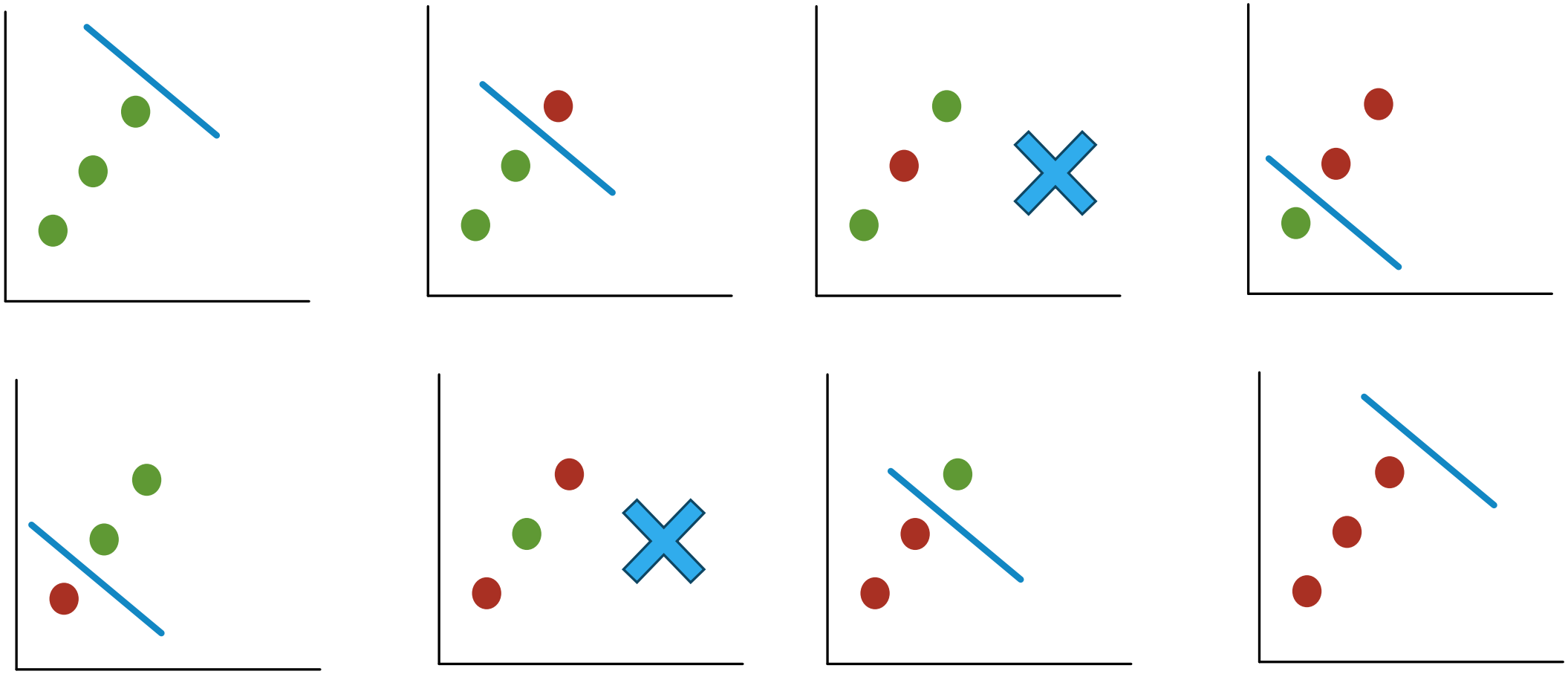


Possible Class Combinations



Can straight lines classify all possible combination of classes for this configuration of 3 points?

Possible Class Combinations



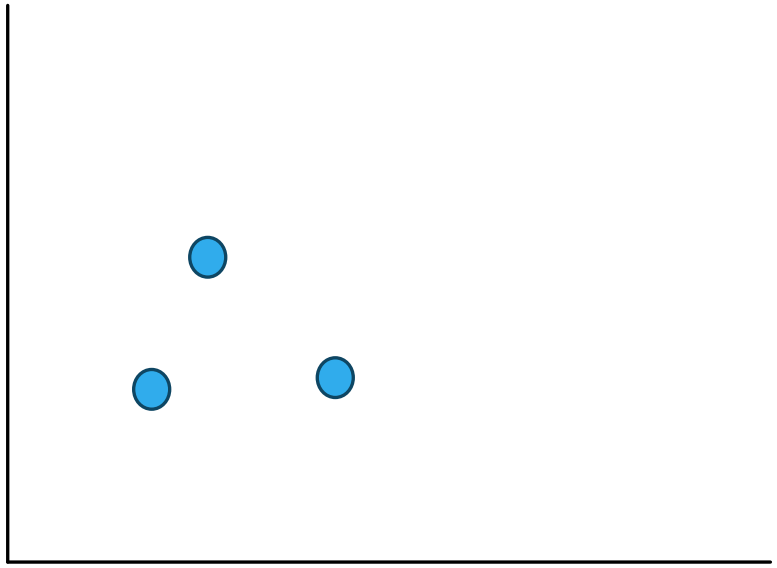
Can straight lines classify all possible combination of classes for this configuration of 3 points?

No

Three Points

Can straight lines classify all possible combination of classes for any configuration of 3 points?

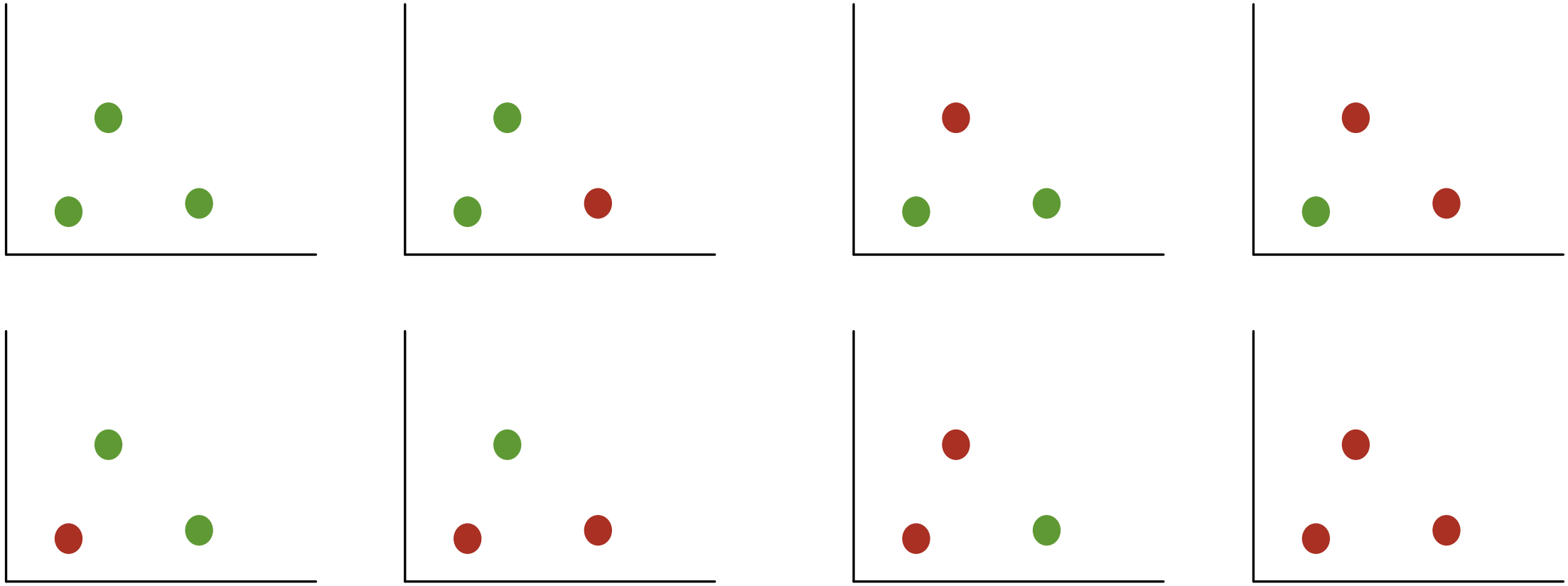
Three Points



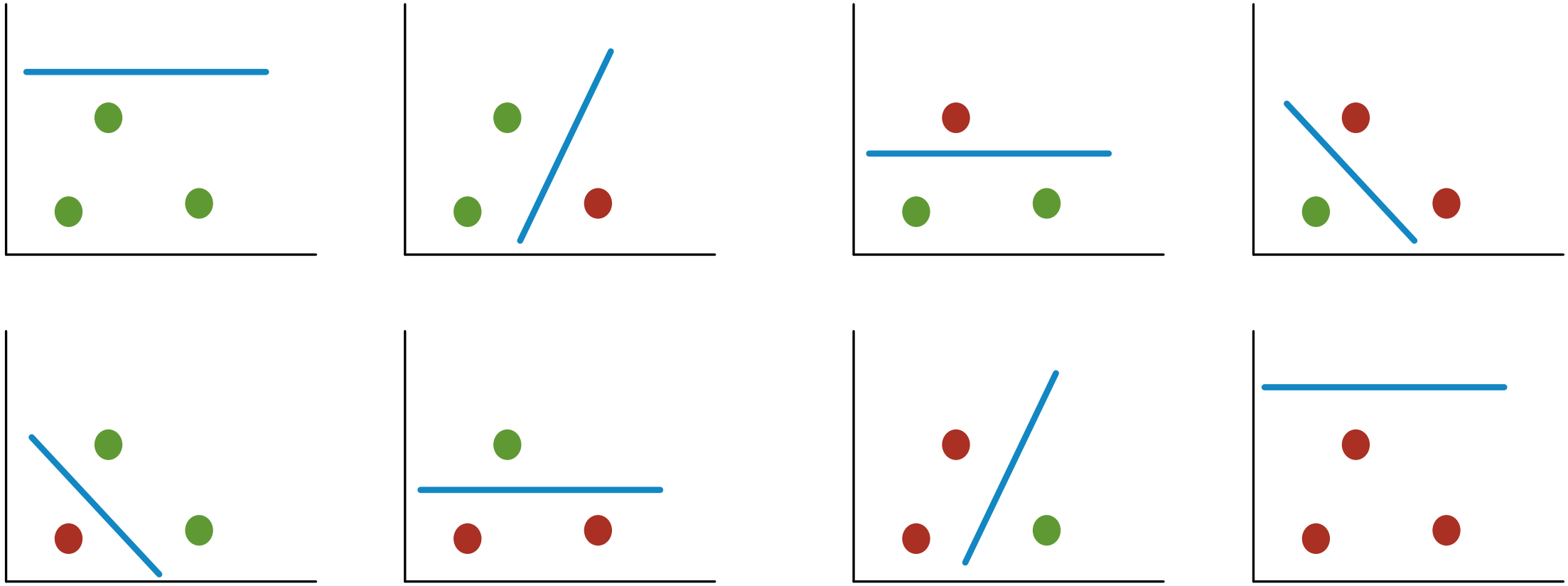
Can straight line classify all possible combination of classes for this configuration of 3 points?

Let’s see for this configuration of three points

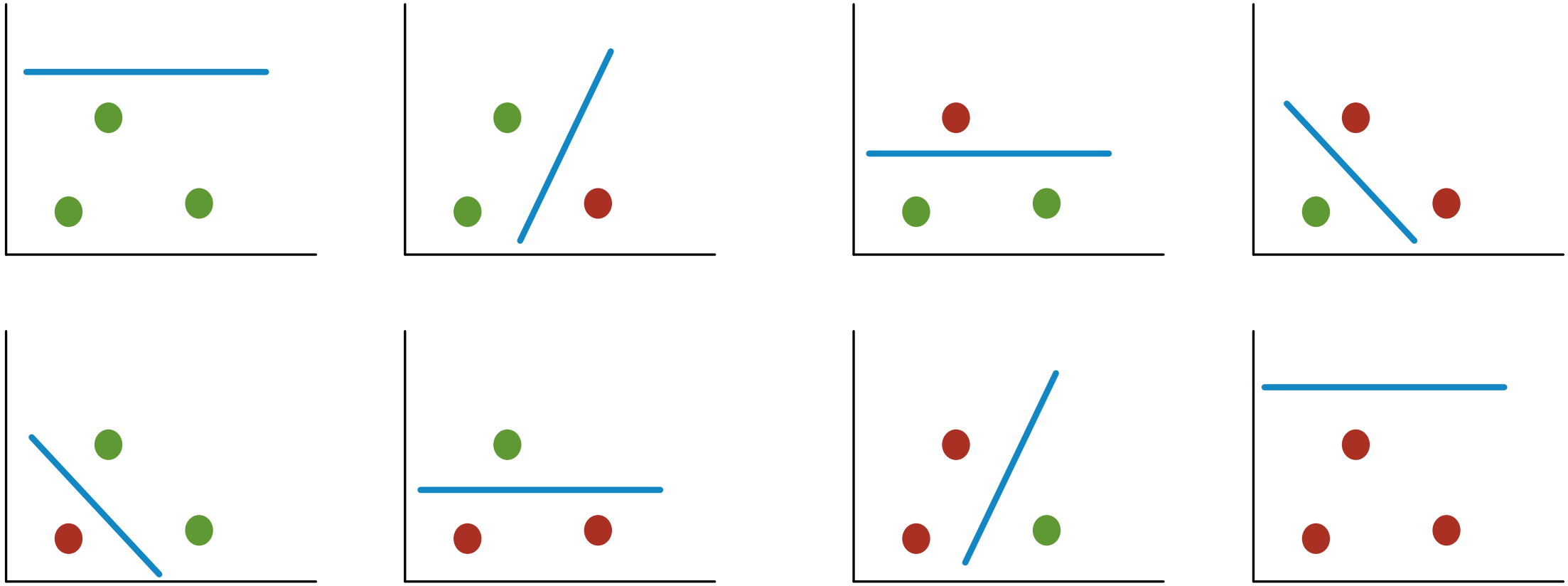
Three Points



Three Points

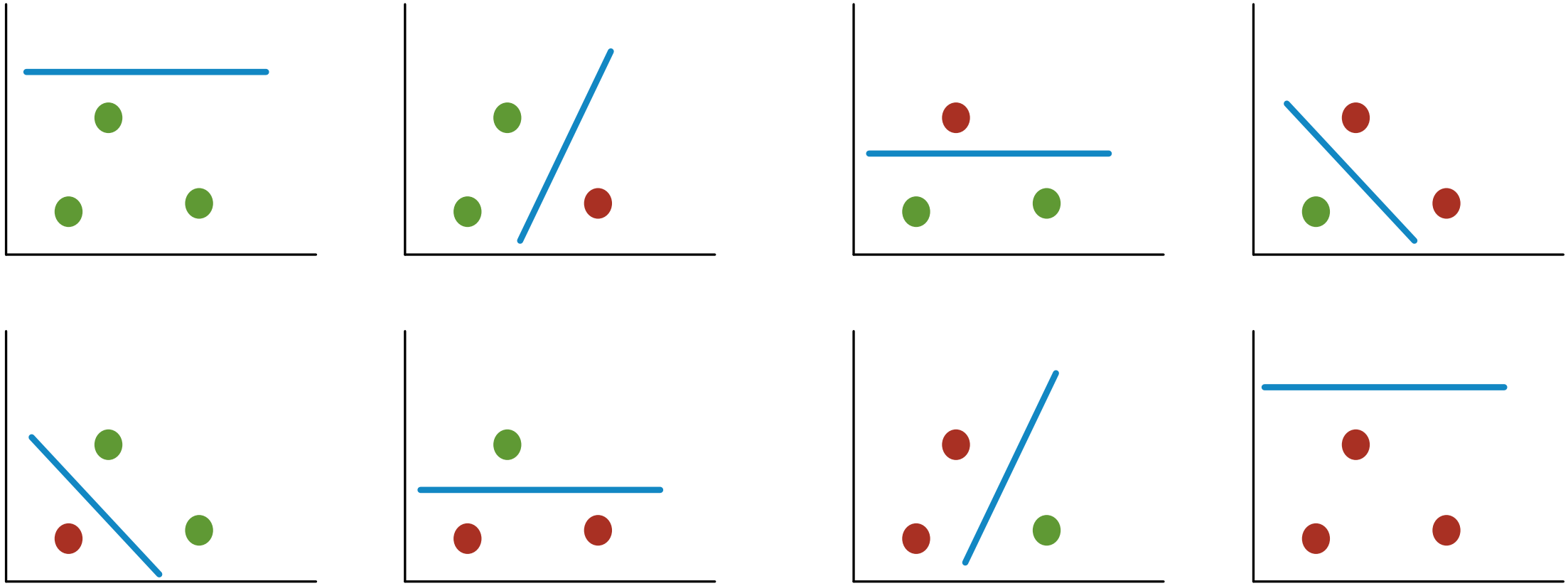


Three Points



So, there is at least one configuration of 3 points in 2D plane for which straight lines can classify all possible combination of classes

Three Points

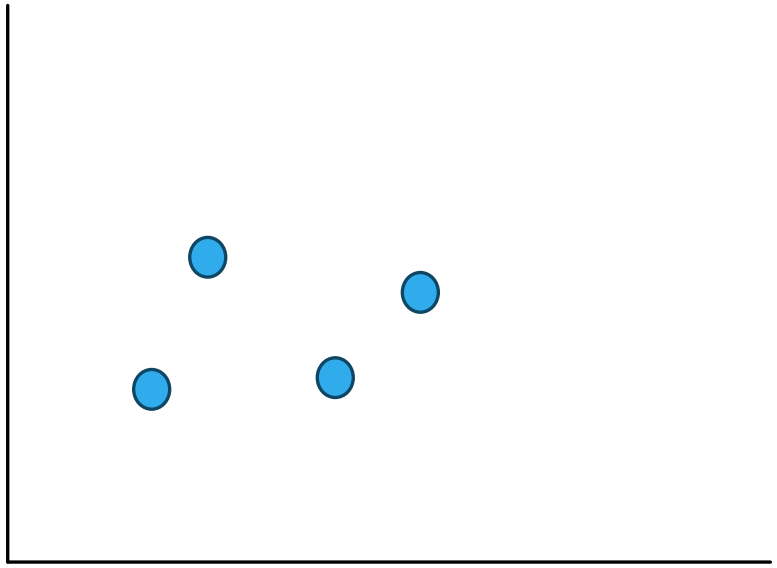


So, we say straight line can shatter 3 points on 2D plane

Shattering

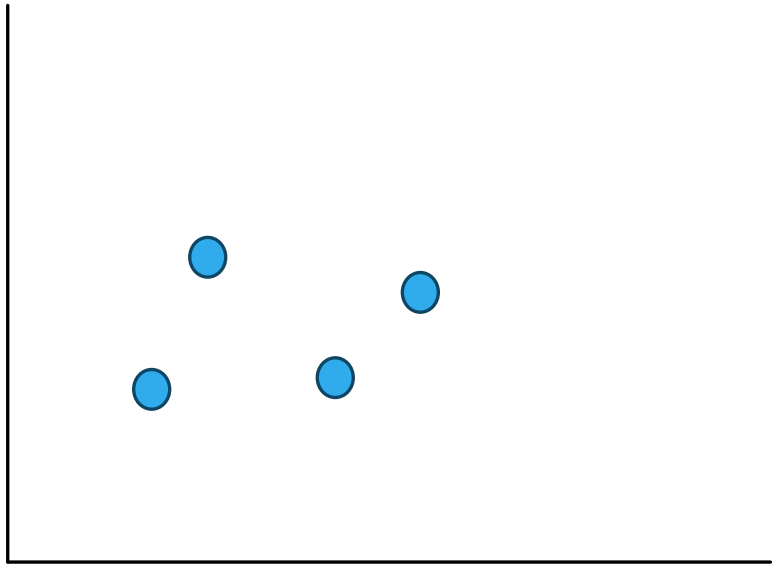
A set of N points is said to be shattered by a hypothesis space H if there are hypotheses in H that can separate the positive and negative examples in all of the 2^N possible ways for at least one configuration of data points

Four Points



Can straight lines shatter 4 points in 2D Plane?

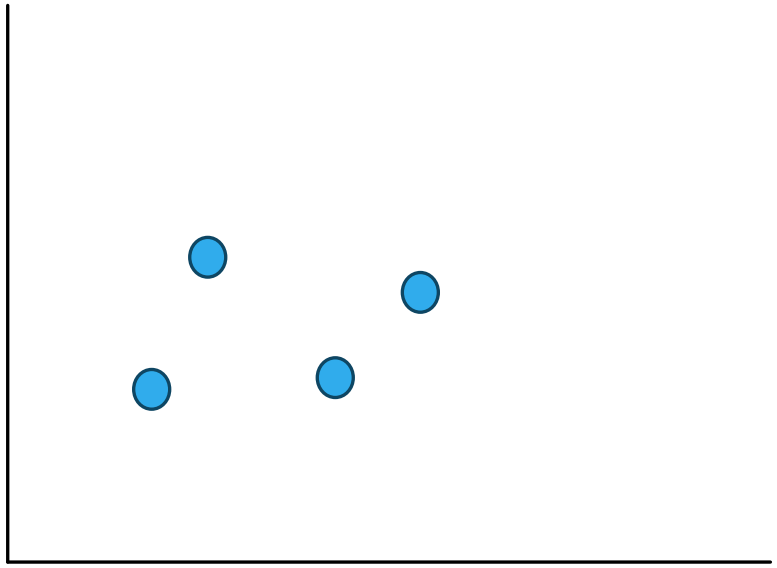
Four Points



Can straights line shatter any 4 points in 2D Plane?

No

Vapnik–Chervonenkis (VC)-Dimension



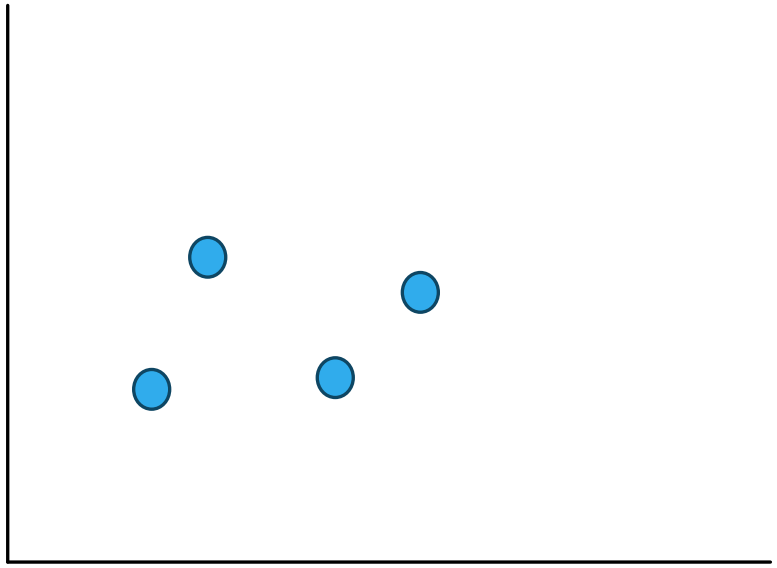
Can straight lines shatter any 4 points in 2D Plane?

No

That means the maximum number of points that can be shattered by straight lines in 2D plane is 3

We say that the VC-Dimension of the hypothesis space consisting of straight lines in 2D plane is 3

VC-Dimension

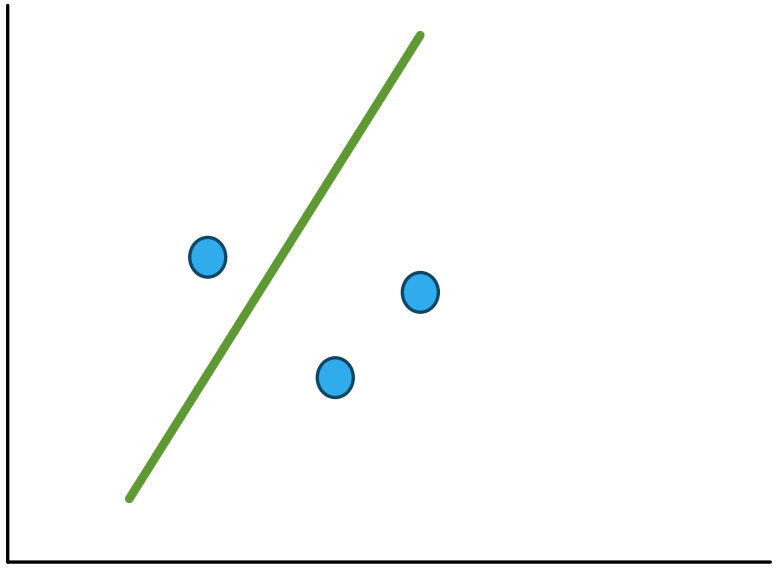


The maximum number of points that can be shattered by a hypothesis space is called the VC-Dimension of that Hypothesis space

VC-Dimension characterizes the expressive power or capacity of a hypothesis space

If a classification model can shatter at most *D* points by changing its parameters, we say that the VC-dimension of the classification model is *D*

VC-Dimension



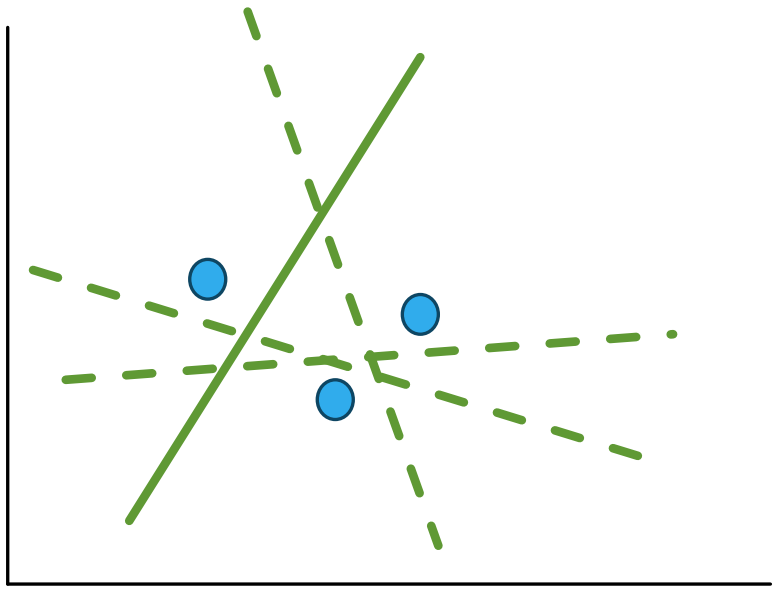
If a classification model can shatter at most D points by changing its parameters, we say that the VC-dimension of the classification model is D

Consider a classification model that creates a decision boundary

$$y = mx + c$$

Where $m\ and\ c$ are parameters of the model

VC-Dimension



If a classification model can shatter at most D points by changing its parameters, we say that the VC-dimension of the classification model is D

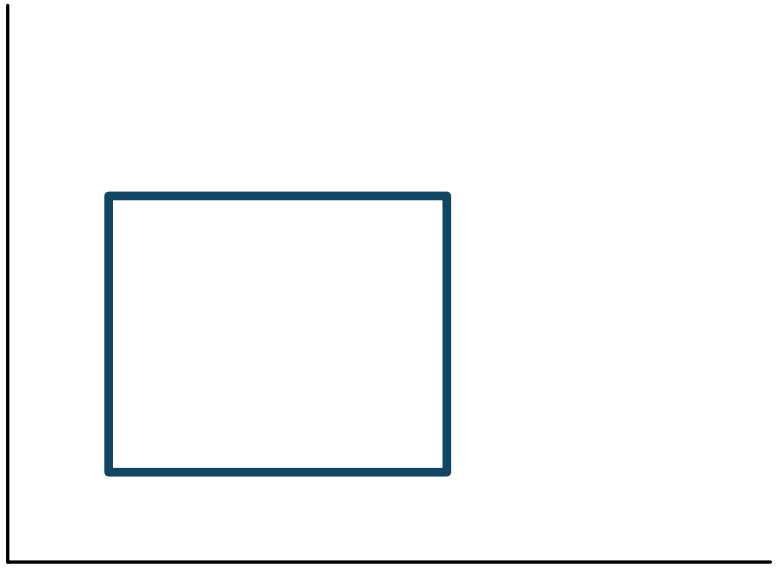
Consider a classification model that creates a decision boundary

$$y = mx + c$$

Where $m\ and\ c$ are parameters of the model

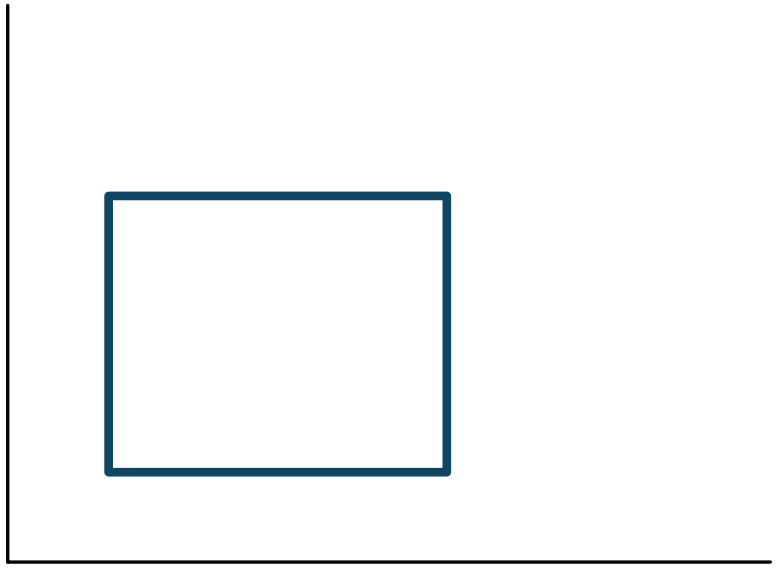
Then, by suitably changing $m\ and\ c$, we can create a hypothesis space consisting of all possible straight lines

VC-Dimension



What is the VC-Dimension of the hypothesis space consisting of axis-aligned rectangles?

VC-Dimension



What is the VC-Dimension of the hypothesis class of axis-aligned rectangles?

Answer: 4

VC-Dimension & Test Error

Consider a classification model with VC-dimension D

Let N be the number of training data. Then,

$$P\left(\textbf{test error} < \textbf{training error} + \sqrt{\frac{1}{N}\left[D\left(\log\left(\frac{2N}{D}\right)+1\right)-\log\left(\frac{\eta}{4}\right)\right]}\right) = 1-\eta$$

$$0 \leq \eta \leq 1$$

This is applicable when $D \ll N$, i.e., number of training samples is way higher than the VC-dimension

VC-Dimension & Test Error

Consider a classification model with VC-dimension D

Let N be the number of training data. Then,

$$P\left(\textit{test error} < \textit{training error} + \sqrt{\frac{1}{N}\left[D\left(\log\left(\frac{2N}{D}\right)+1\right)-\log\left(\frac{\eta}{4}\right)\right]}\right) = 1-\eta$$

$$0 \leq \eta \leq 1$$

This is applicable when $D \ll N$, i.e., number of training samples is way higher than the VC-dimension

But if VC-dimension is high, the model is more expressive, i.e., more complex, that means more chance of overfitting, and so, higher test error is possible

Preparing Data for ML

Challenges in Data for ML

- Good ML models can’t be designed without good quality data
- Most practical data come with various problems
  - Noise
  - Missing data
  - Duplicate data
  - Inconsistent data, etc.

Challenges in Data for ML

- Data Cleaning
  - Reducing noise
  - Handling duplicate data
  - Handling inconsistent data
  - Handling missing values
  - May require domain knowledge to deal with these problems

Student ID	Student Name	Age	GPA	Classification
100122014	Joseph	21	3.5	Junior
100232015	Patrick	200	3.2	Sophomore
100122012	Seller	24	3.0	Senior
100342013	Roger	23	234	Senior
100942012	Davis	2.8	3.7	Sophomore
	Travis	23	3.4	Sr
100982015	Alex	27		Sophomore
100982013	Trevor	-22	4.0	Senior
AUC2016XC	Aman	30	3.5	Jr

Missing Data

Inconsistent Data

Noisy Data

https://medium.com/@casternxavier/get-rid-of-the-dirt-from-your-data-data-cleaning-techniques-743253b1c3e5

How to Clean Data

- Dealing with missing values
  - Data deletion: results in loss of data
  - Imputation: through informed guess, or by using regression
- Dealing with noise
  - Binning
    - Divide the total data into bins. Replace the value of the noise data based on some characteristics of the bin (such as mean)
  - Regression
  - Outlier analysis

Scale of Features

- Suppose, I want to find the following abnormalities
  - Hemoglobin disorder (based on Hb %)
  - Kidney diseases (based on creatinine)

- In my samples
  - Average Hb is 13500 mg/dL
  - Average creatine is 0.7 mg/dL

Scale of Features

- Suppose, I want to find the following abnormalities
  - Hemoglobin disorder (based on Hb %)
  - Kidney diseases (based on creatinine)

- I assume the following normal levels
  - Hb: 13500 mg/dL
  - Creatine: 0.7 mg/dL

- Suppose, I find a person with
  - Hb: 12500 mg/dL
  - Creatinine: 1.7 mg/dL

Scale of Features

- Suppose, I want to find the following abnormalities
  - Hemoglobin disorder (based on Hb %)
  - Kidney diseases (based on creatinine)
- I assume the following normal levels
  - Hb: 13500 mg/dL
  - Creatine: 0.7 mg/dL
- Suppose, I find a person with
  - Hb: 12500 mg/dL
  - Creatinine: 1.7 mg/dL
- If I just go by absolute value of the numbers, deviations from normal
  - Hb: 1000 mg/dL
  - Creatine: 1 mg/dL

What should I conclude?

Scale of Features

- Suppose, I want to find the following abnormalities
  - Hemoglobin disorder (based on Hb %)
  - Kidney diseases (based on creatinine)

- I assume the following normal levels
  - Hb: 13500 mg/dL
  - Creatine: 0.7 mg/dL

- Suppose, I find a person with
  - Hb: 12500 mg/dL
  - Creatinine: 1.7 mg/dL

- If I just go by absolute value of the numbers, deviations from normal
  - Hb: 1000 mg/dL
  - Creatine: 1 mg/dL

What should I conclude?

You may be tempted to conclude that the person has Hb disorder because there is a huge change in Hb compared to change in Creatinine

Scale of Features

- Suppose, I want to find the following abnormalities
  - Hemoglobin disorder (based on Hb %)
  - Kidney diseases (based on creatinine)
- I assume the following normal levels
  - Hb: 13500 mg/dL
  - Creatine: 0.7 mg/dL
- Suppose, I find a person with
  - Hb: 12500 mg/dL
  - Creatinine: 1.7 mg/dL
- If I just go by absolute value of the numbers, deviations from normal
  - Hb: 1000 mg/dL
  - Creatine: 1 mg/dL

What should I conclude?

You may be tempted to conclude that the person has Hb disorder because there is a huge change in Hb compared to change in Creatinine

But this conclusion may be wrong

This wrong conclusion may happen because the normal values are at different range

Scale of Features

- Suppose, I want to find the following abnormalities
  - Hemoglobin disorder (based on Hb %)
  - Kidney diseases (based on creatinine)
- I assume the following normal levels
  - Hb: 13500 mg/dL
  - Creatine: 0.7 mg/dL
- Suppose, I find a person with
  - Hb: 12500 mg/dL
  - Creatinine: 1.7 mg/dL
- If I just go by absolute value of the numbers, deviations from normal
  - Hb: 1000 mg/dL
  - Creatine: 1 mg/dL

What should I conclude?

You may be tempted to conclude that the person has Hb disorder because there is a huge change in Hb compared to change in Creatinine

But this conclusion may be wrong

This wrong conclusion may happen because the normal values are at different range

Same may happen for ML models

Scale of Features

- Suppose, I want to find the following abnormalities
  - Hemoglobin disorder (based on Hb %)
  - Kidney diseases (based on creatinine)
- I assume the following normal levels
  - Hb: 13500 mg/dL
  - Creatine: 0.7 mg/dL
- Suppose, I find a person with
  - Hb: 12500 mg/dL
  - Creatinine: 1.7 mg/dL
- If I just go by absolute value of the numbers, deviations from normal
  - Hb: 1000 mg/dL
  - Creatine: 1 mg/dL

What should I conclude?

You may be tempted to conclude that the person has Hb disorder because there is a huge change in Hb compared to change in Creatinine

But this conclusion may be wrong

This wrong conclusion may happen because the normal values are at different range

Same may happen for ML models

So, want to bring the different features at the same range

Scale of Features

- If we do not normalize, features with lower value range may loose their meaning when combined response is created

- Large spread in feature value may mislead the ML model

Data Standardization

- Standardization at input: create feature values with 0 mean and unit variance
- Suppose, there are n data points
- For a particular feature f, the feature values for n data points are $f_1, f_2, \dots, f_n$
- The standardized feature value is

$$f_i' = \frac{f_i - \mu_f}{\sigma_f}$$

$$\mu_f : Mean\ of\ f_1, f_2, \dots, f_n$$

$$\sigma_f : std\ of\ f_1, f_2, \dots, f_n$$

aka z-score normalization

Min-max Normalization

- Suppose, there are n data points
- For a particular feature f, the feature values for n data points are $f_1, f_2, \dots, f_n$
- The normalized feature value is

$$f_i' = \frac{f_i - \min(f_1, f_2, \dots, f_n)}{max(f_1, f_2, \dots, f_n) - \min(f_1, f_2, \dots, f_n)}$$

Linear Regression

Regression

- Experience

Data

https://www.pexels.com/

Data

Regression

- Experience

Data

Price: 3.4 Cr

Price: 1.8 Cr

Price: 1.2 Cr

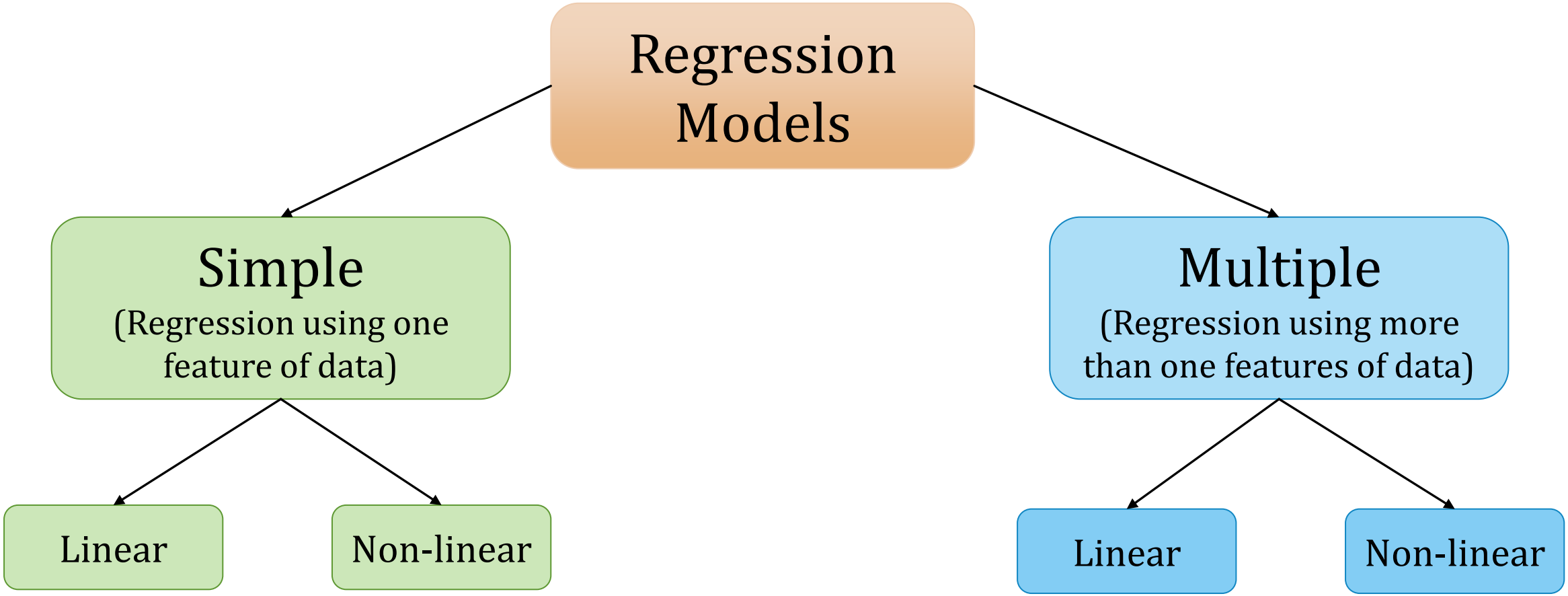
Label

Regression

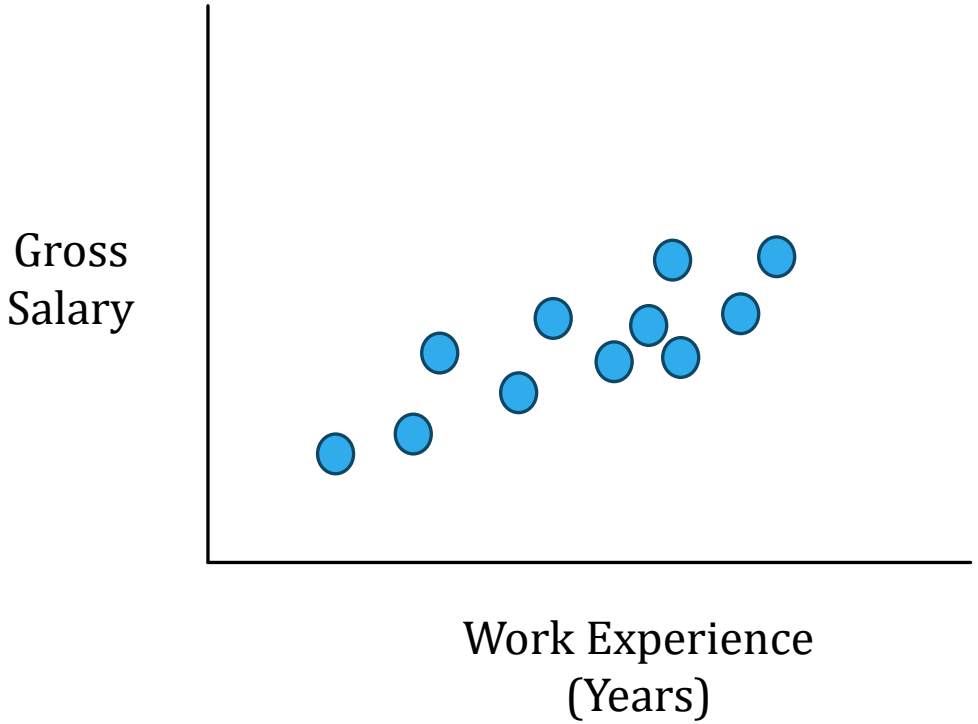
- Task: Predict the price of this house

https://www.pexels.com/

Regression

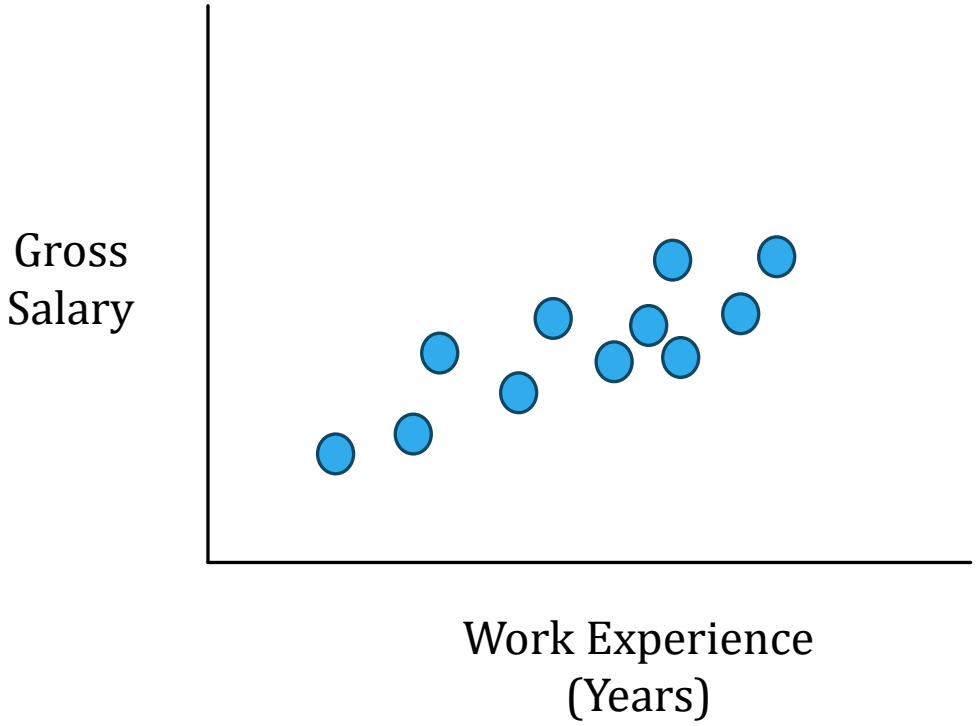

Regression Models
Simple
(Regression using one feature of data)
Multiple
(Regression using more than one features of data)
Linear
Non-linear
Linear
Non-linear

Regression



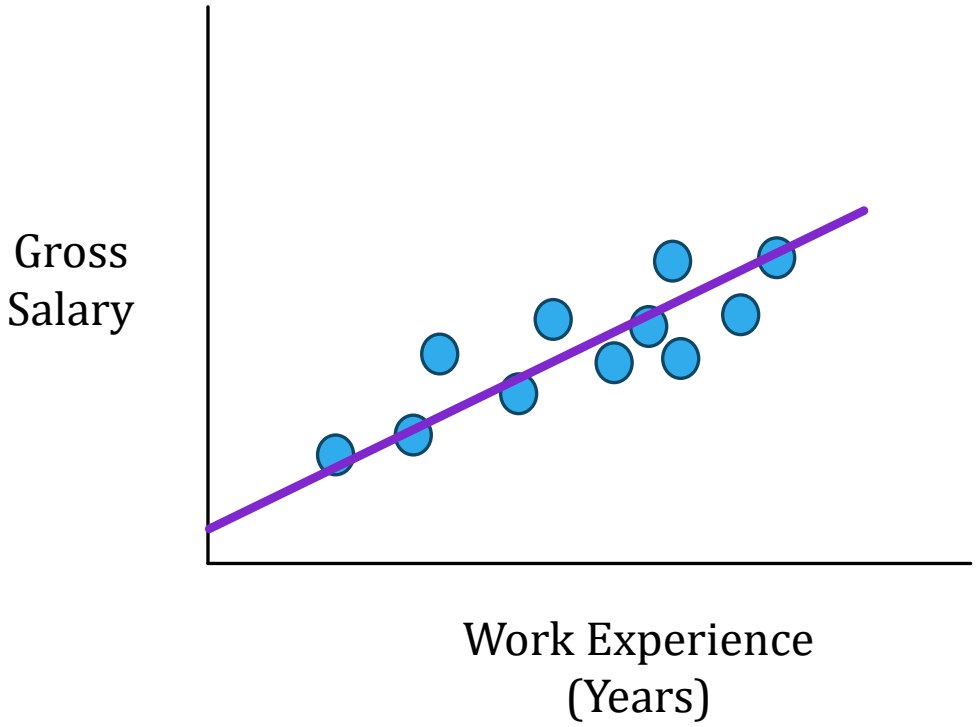
Training data

Regression



Which curve would you like to fit for regression?

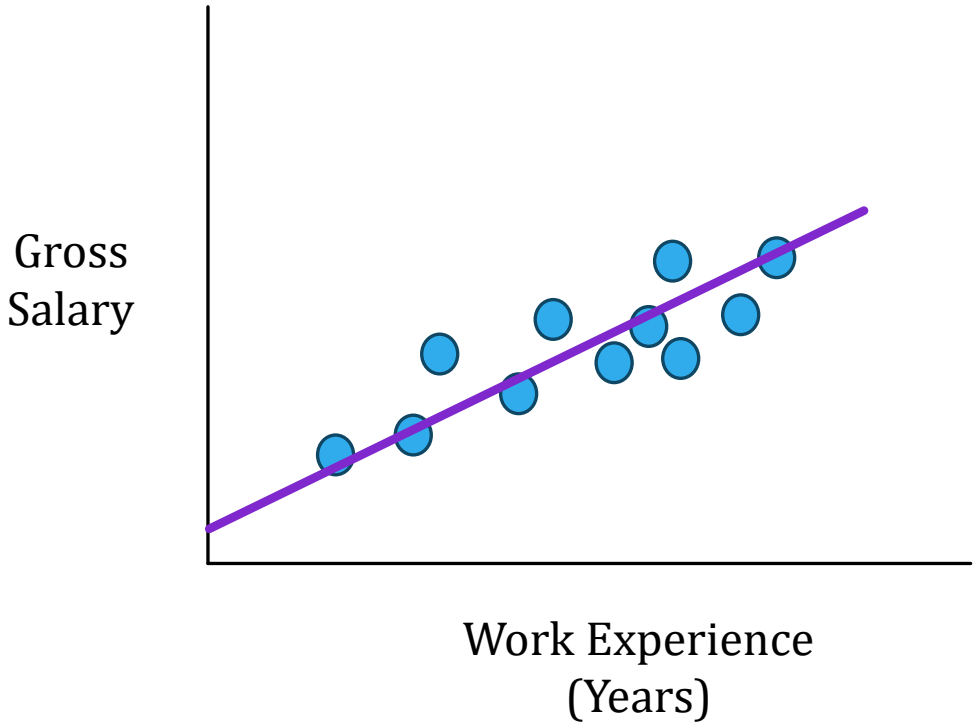
Regression



Which curve would you like to fit for regression?

A straight line

Regression

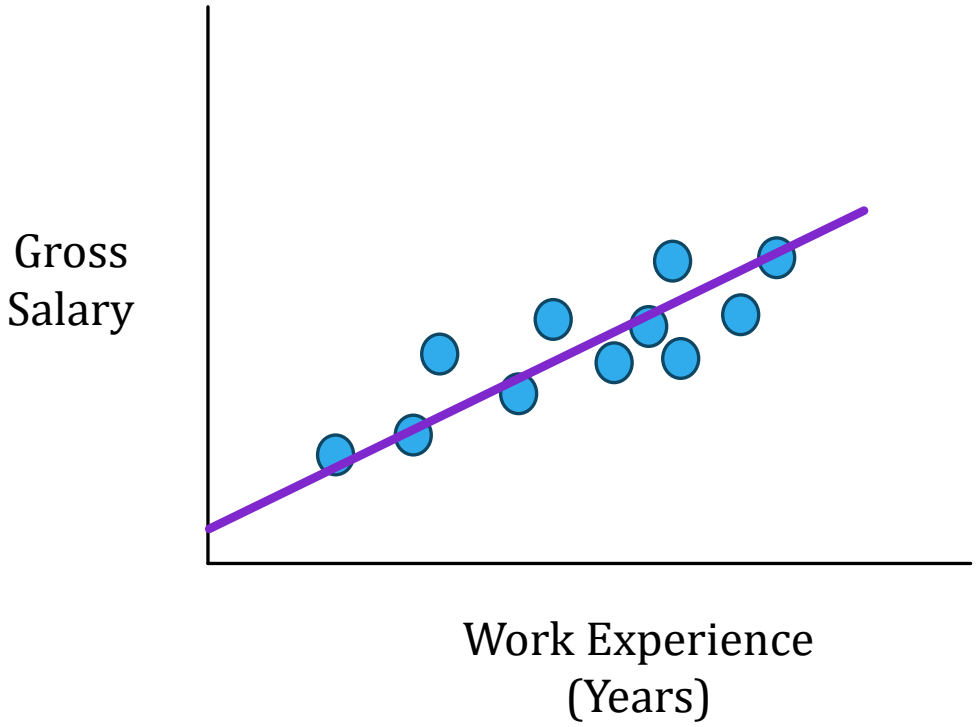


Which curve would you like to fit for regression?

A straight line

Immediately, you made an assumption that my data can be explained using straight line

Regression



Which curve would you like to fit for regression?

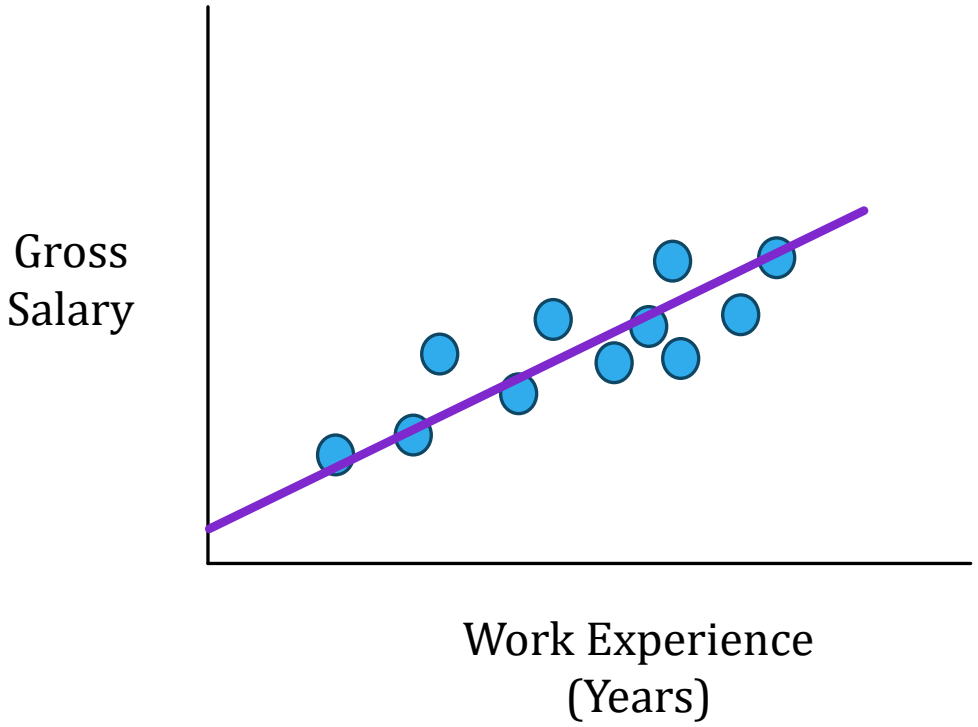
A straight line

Immediately, you made an assumption that my data can be explained using straight line

Inductive bias

Which type of inductive bias?

Regression



Which curve would you like to fit for regression?

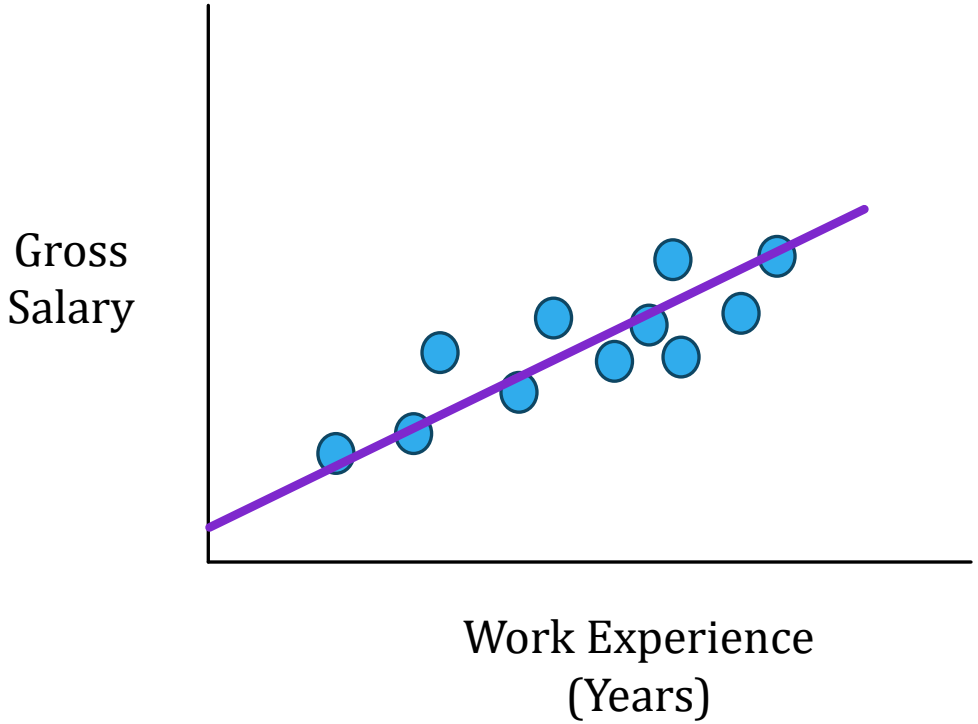
A straight line

Immediately, you made an assumption that my data can be explained using straight line

Inductive bias

Which type of inductive bias?
Restriction bias

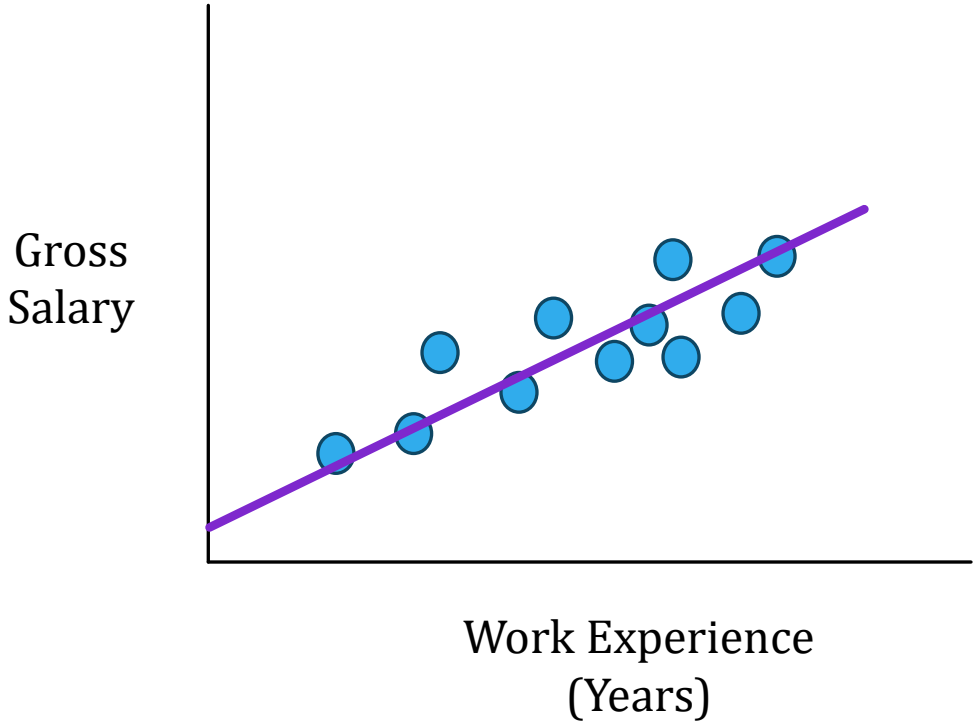
Regression



Equation of the straight line

$$y = mx + c$$

Regression

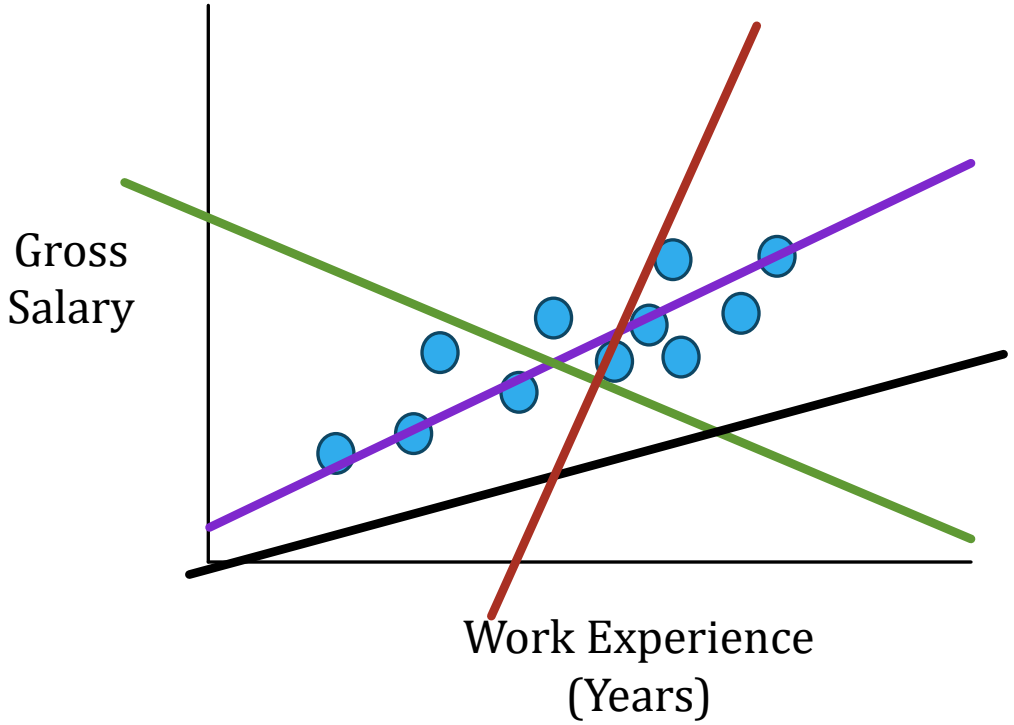


Equation of the straight line

$$y = mx + c$$

Parameters of the equation/regression model

Regression



Equation of the straight line

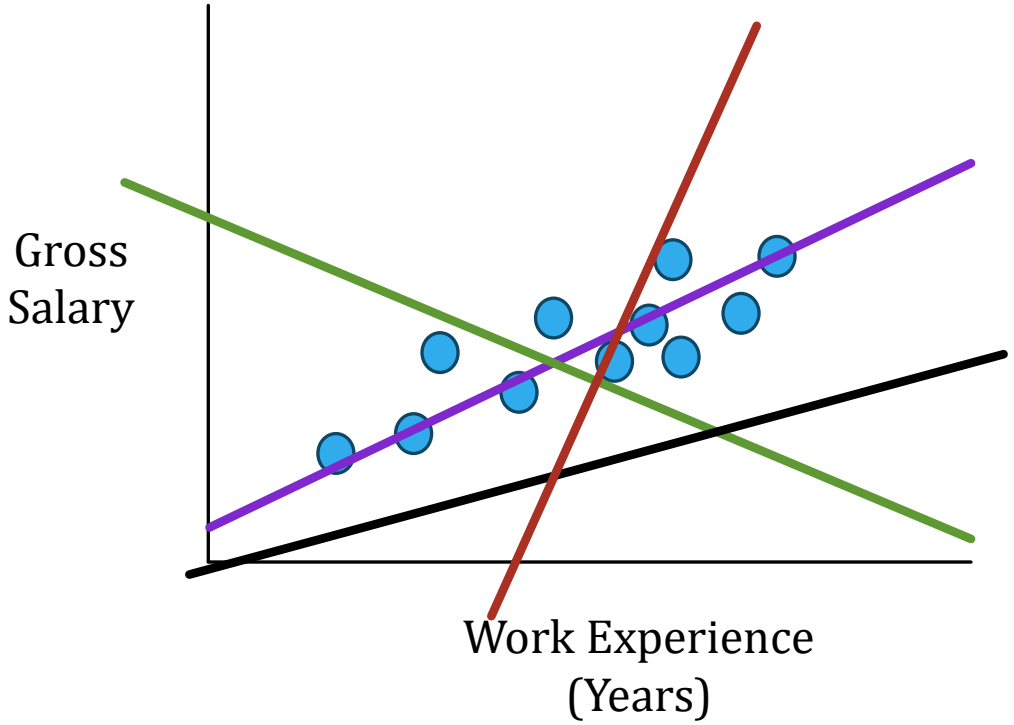
$$y = mx + c$$

Parameters of the equation/regression model

For different values of parameters (m *and* c), I am going to get different straight lines

Which straight line am I going to prefer?

Regression



Equation of the straight line

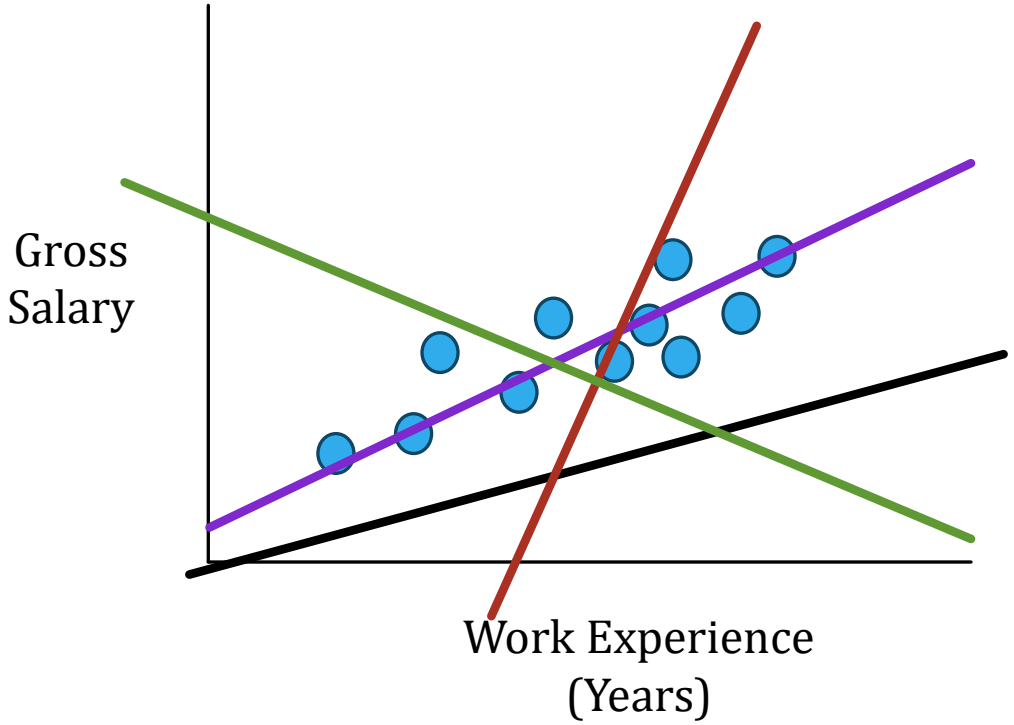
$$y = mx + c$$

Parameters of the equation/regression model

For different values of parameters (m *and* c), I am going to get different straight lines

Which straight line am I going to prefer? Use error

Regression



Equation of the straight line

$$y = mx + c$$

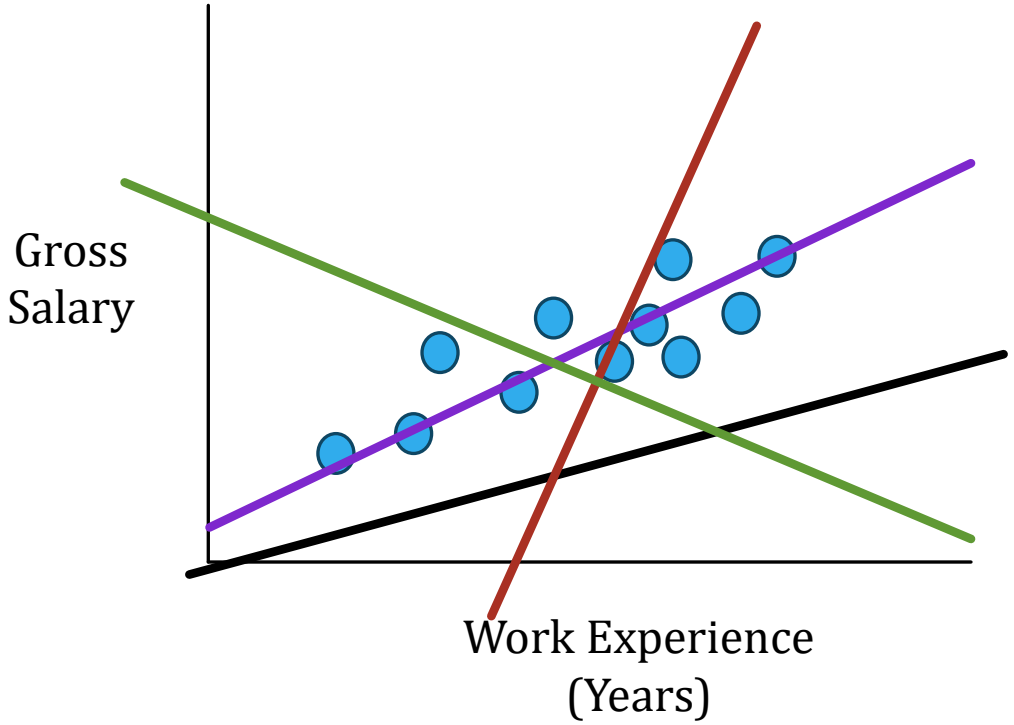
Parameters of the equation/regression model

For different values of parameters (m *and* c), I am going to get different straight lines

Which straight line am I going to prefer? **Purple**

Because the purple line has the least total error for training data

Goal of Linear Regression



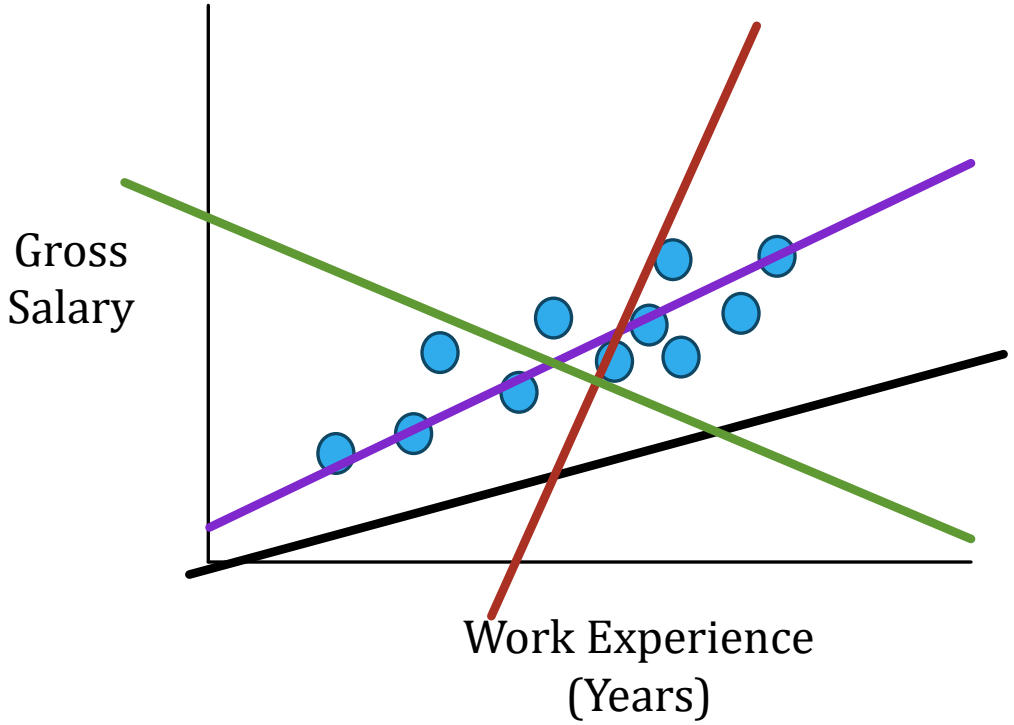
Equation of the straight line

$$y = mx + c$$

Parameters of the equation/regression model

Find a straight line (*an* $\boldsymbol{m}$ ***and*** $\boldsymbol{c}$) that minimizes the total error in training data

Goal of Linear Regression



Equation of the straight line

$$y = mx + c$$

Parameters of the equation/regression model

Find a straight line (an $\boldsymbol{m}$ $\boldsymbol{and}$ $\boldsymbol{c}$) that minimizes the total error in training data

This is a simple linear regression since we are just considering one feature x for regression

Multiple Linear Regression

Equation of the straight line with multiple independent variables

$$y = m_1 x_1 + m_2 x_2 + \cdots + m_p x_p + c$$

Multiple Linear Regression

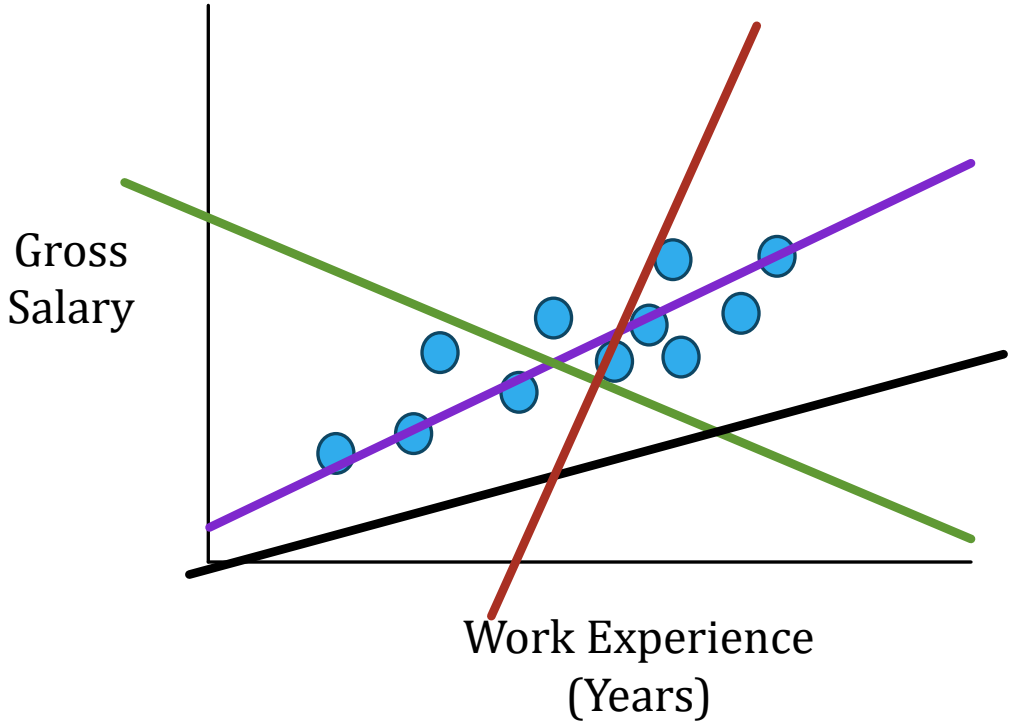
Equation of the straight line

$$\boldsymbol{y = m_1 x_1 + m_2 x_2 + \cdots + m_p x_p + c}$$

Parameters of the equation/regression model

Find a straight line $(an\ \boldsymbol{m_1, m_2, \ldots, m_p\ and\ c})$ that minimizes the total error in training data

Regression



Equation of the straight line

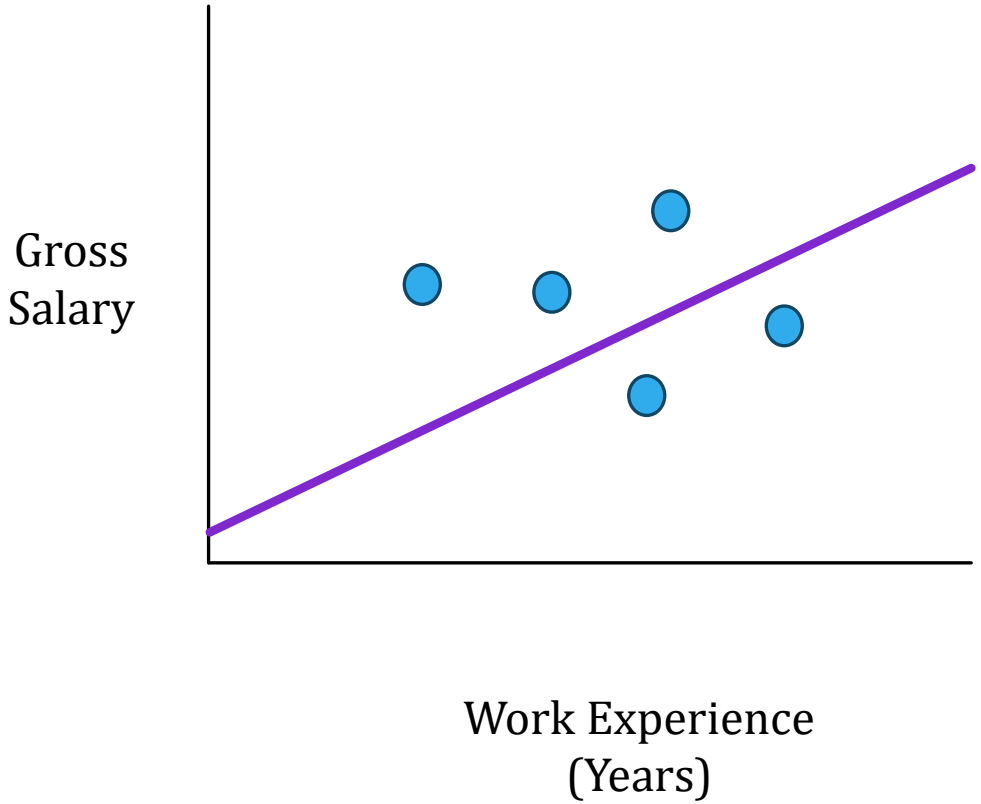
$$y = mx + c$$

Parameters of the equation/regression model

For different values of parameters (m *and* c), I am going to get different straight lines

Which straight line am I going to prefer? Use error

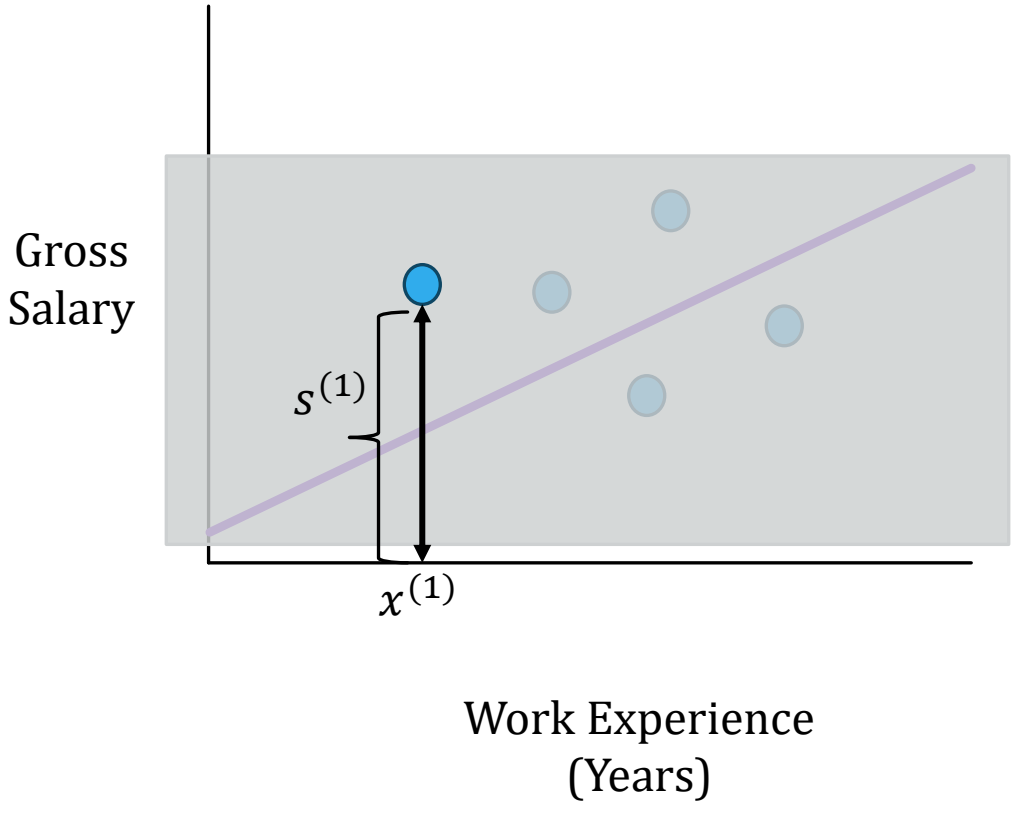
Use of Error



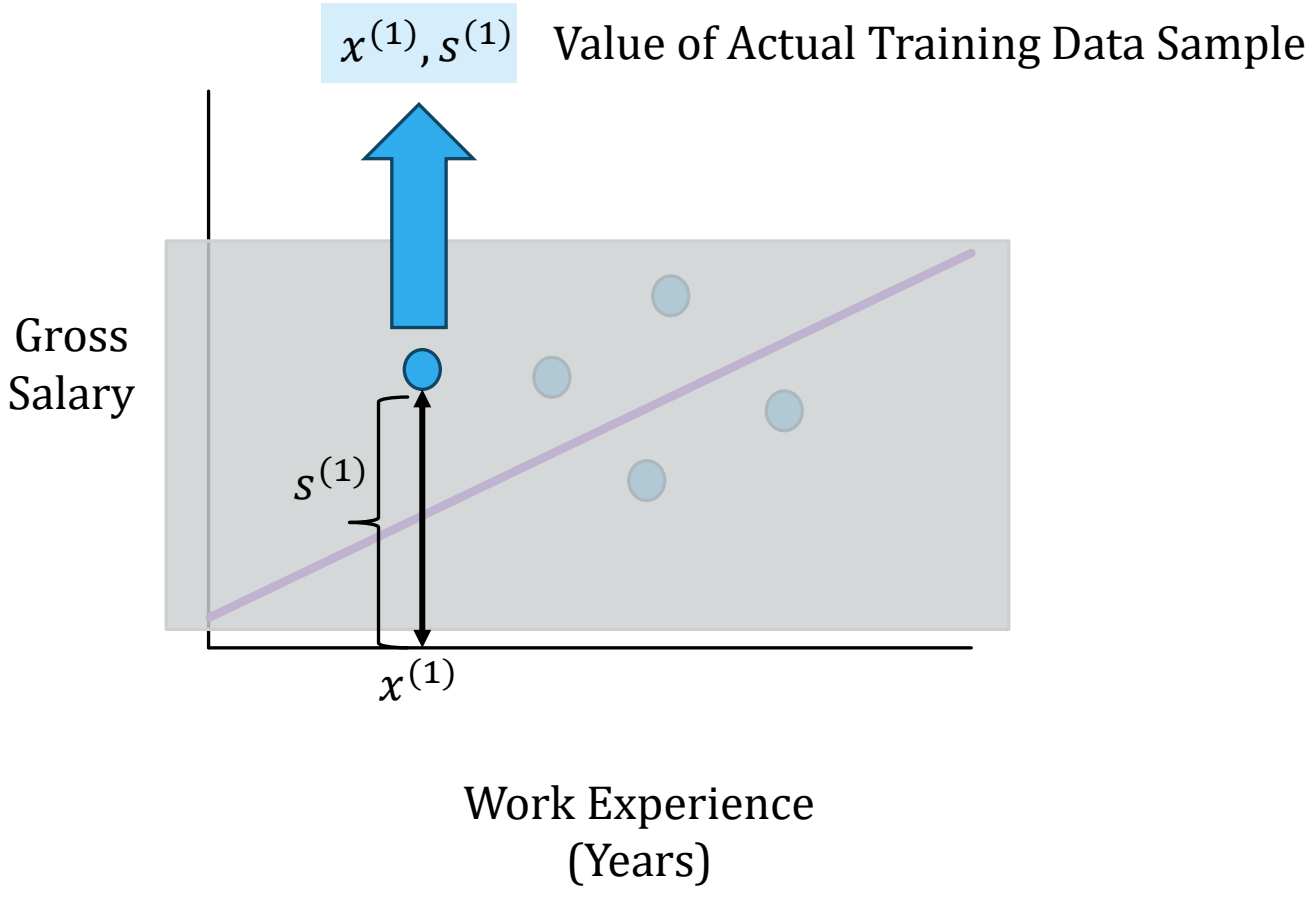
Suppose the blue points are my training data and my model outputs a straight line

$$y = mx + c$$

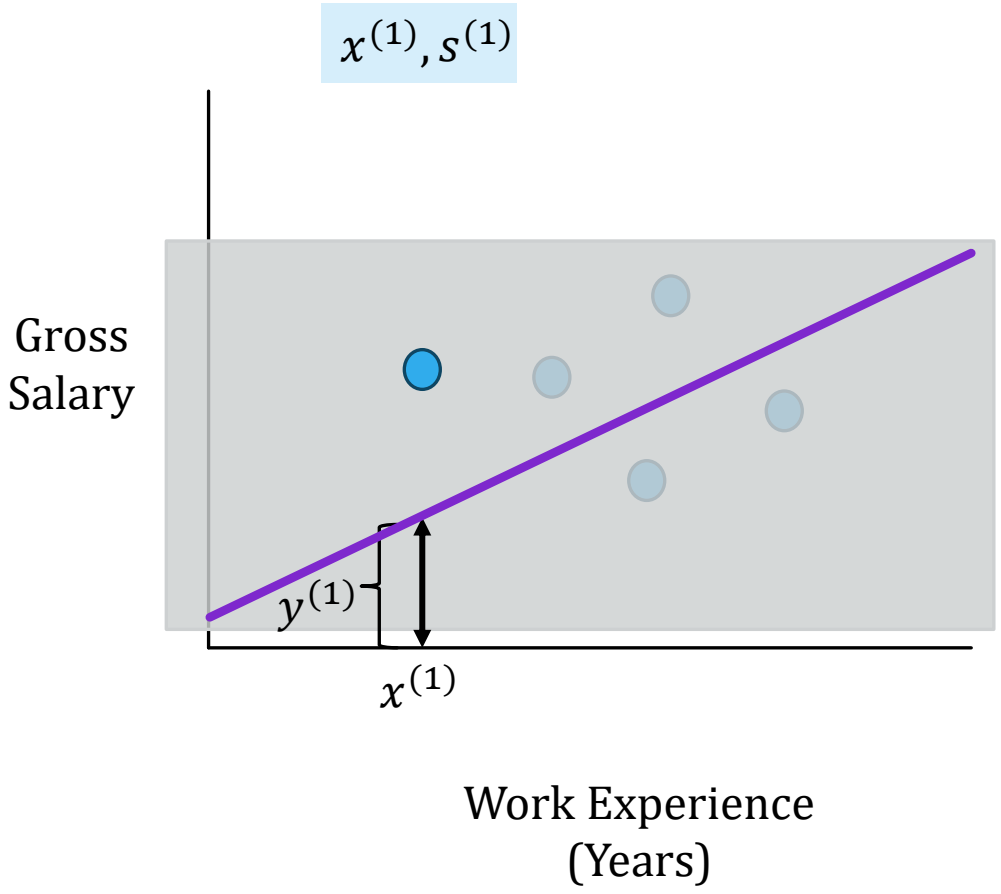
Use of Error



Use of Error

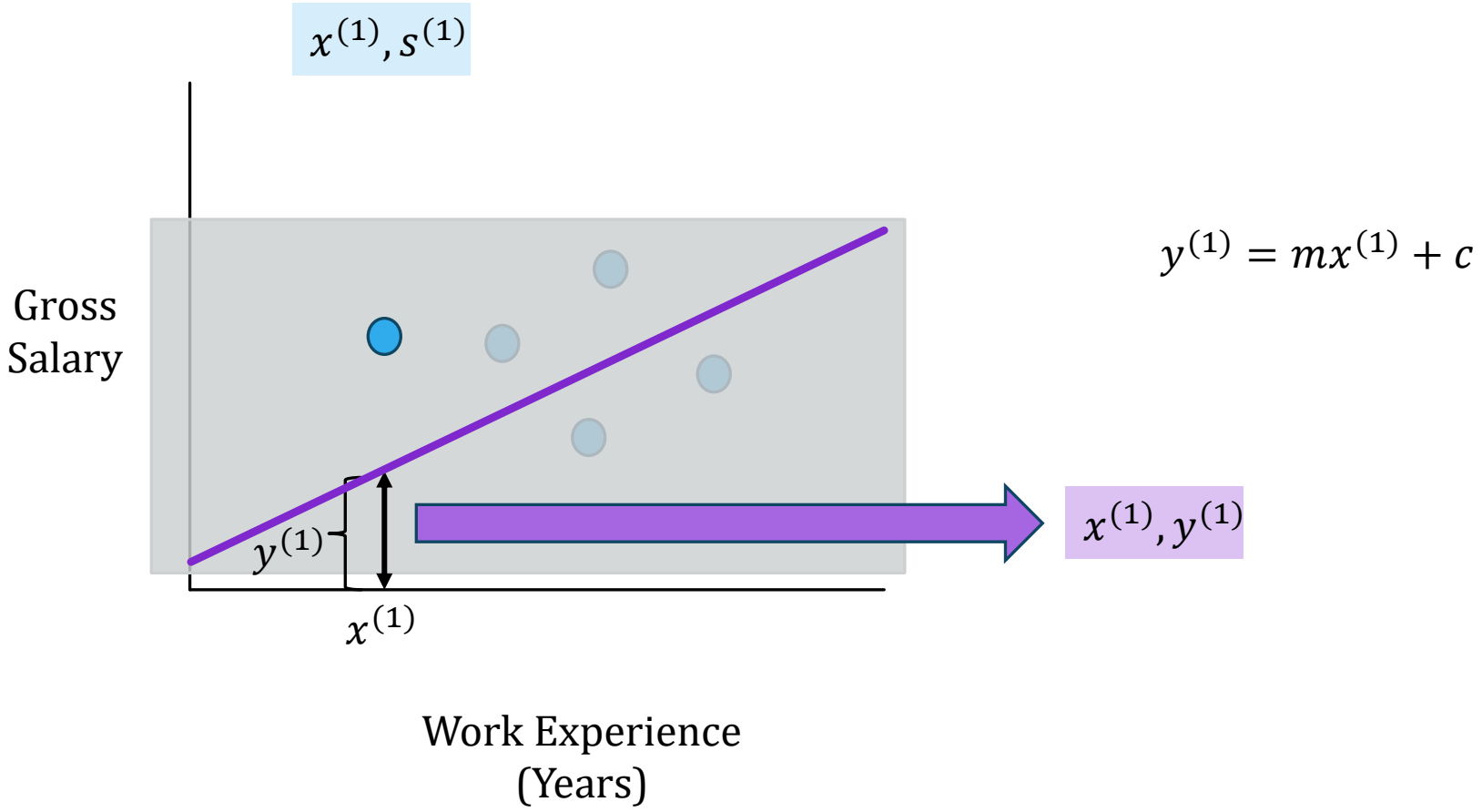

$x^{(1)}, s^{(1)}$
Value of Actual Training Data Sample
Gross Salary
$s^{(1)}$
$x^{(1)}$
Work Experience (Years)

Use of Error



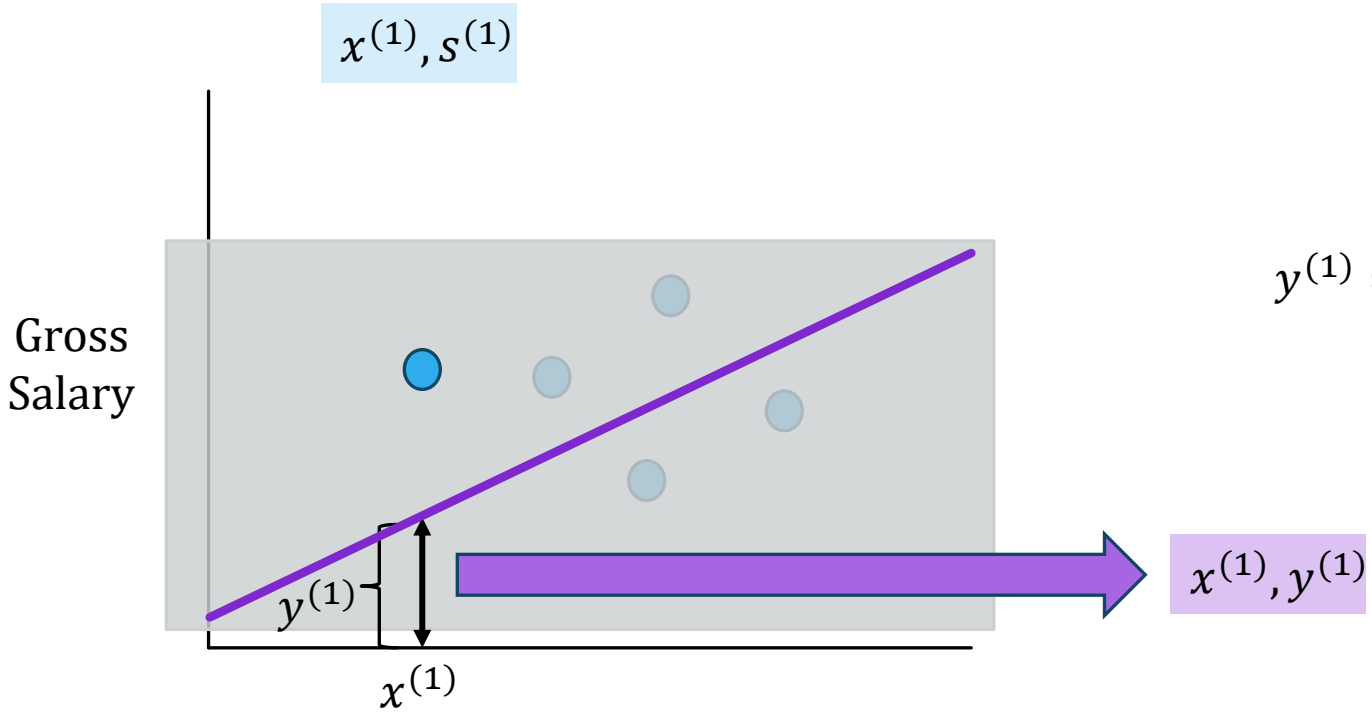
$$y^{(1)} = mx^{(1)} + c$$

Use of Error



$$y^{(1)} = mx^{(1)} + c$$

Use of Error

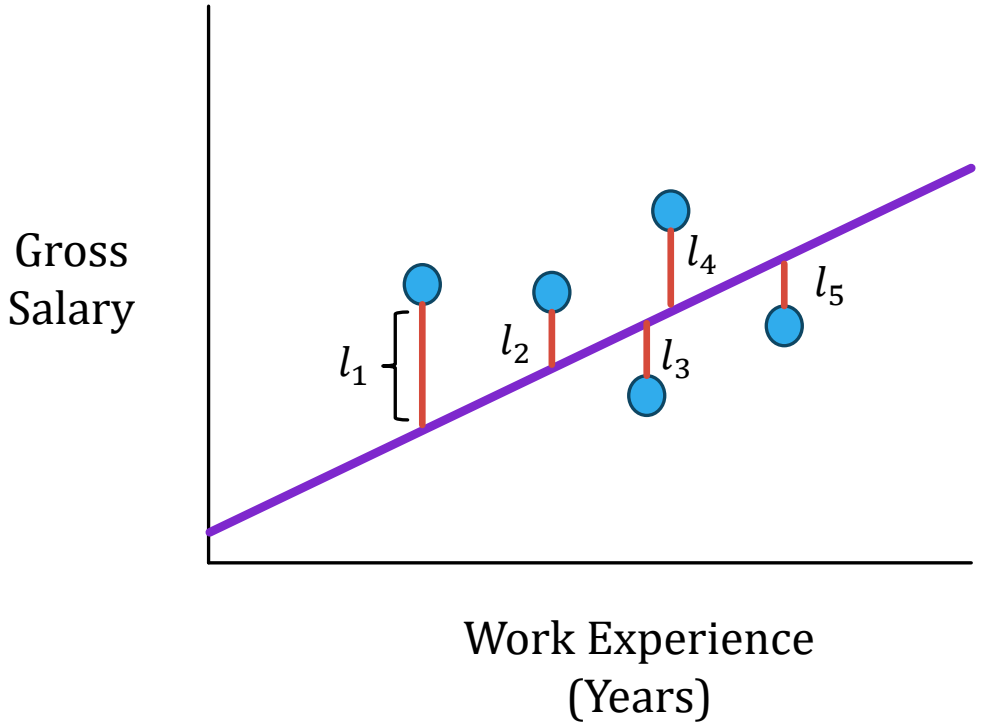


$$y^{(1)} = mx^{(1)} + c$$

Error for this data point w.r.t. prediction

$$e^{(1)} = \left(y^{(1)} - s^{(1)}\right)^2 + \left(x^{(1)} - x^{(1)}\right)^2$$
$$= \left(y^{(1)} - s^{(1)}\right)^2$$

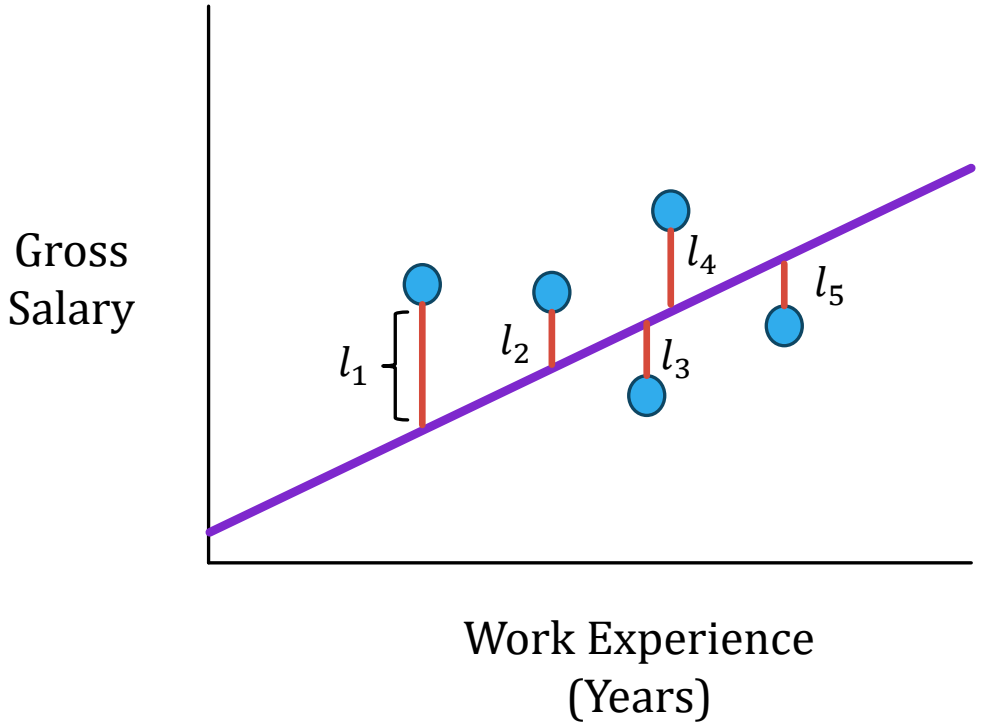
Use of Error



$$e^{(1)} = l_1^2$$
$$e^{(2)} = l_2^2$$

... And so on

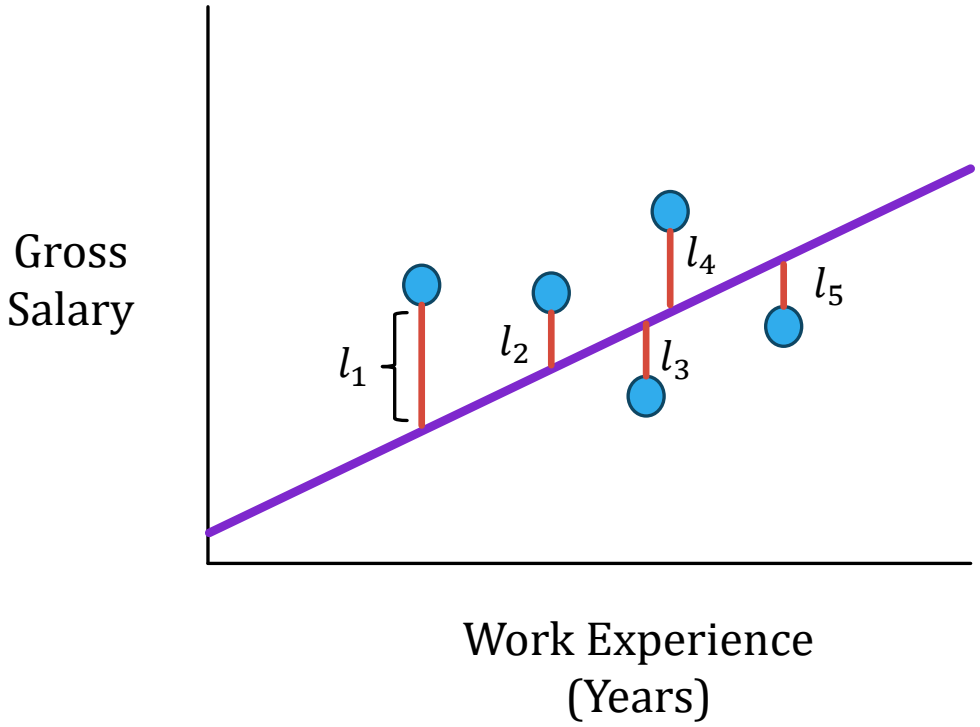
Use of Error



We can calculate error for every training data point and sum them up to get total error

$$E = e^{(1)} + e^{(2)} + \cdots + e^{(N)}$$

Use of Error



We can calculate error for every training data point and sum them up to get total error

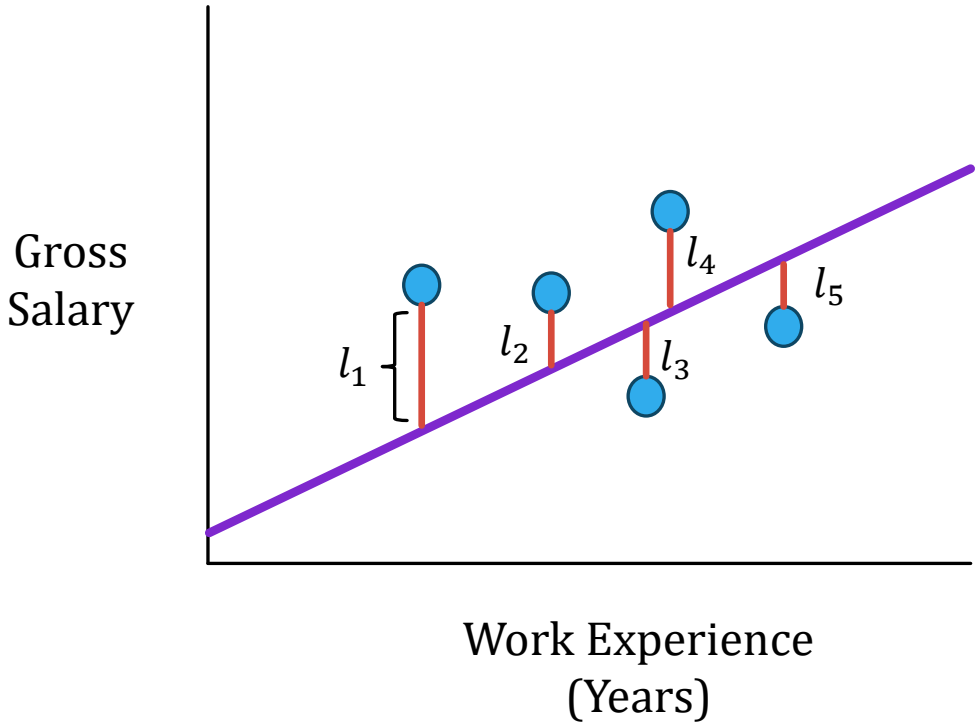
$$E = e^{(1)} + e^{(2)} + \cdots + e^{(N)}$$

$$E = \left(y^{(1)} - s^{(1)}\right)^2 + \left(y^{(2)} - s^{(2)}\right)^2 + \cdots + \left(y^{(N)} - s^{(N)}\right)^2$$

$$E = \sum_{n=1}^{N} \left(y^{(n)} - s^{(n)}\right)^2$$

We want to find a hypothesis (straight line) such that error E is minimized

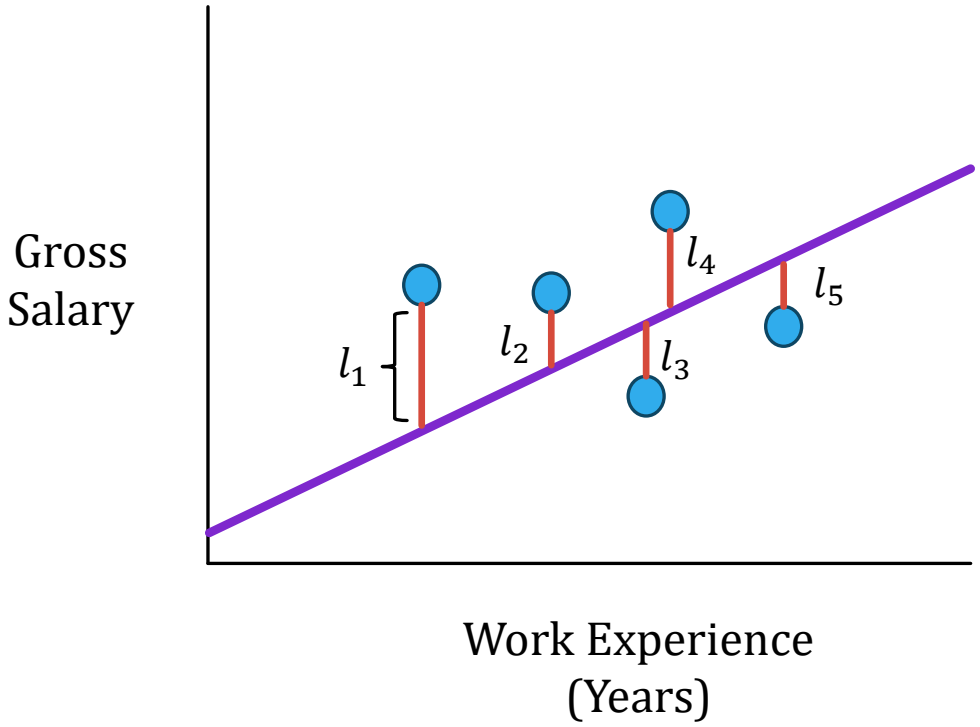
The Objective Function



$$J(m,c) = \frac{1}{2}\sum_{n=1}^{N}\left(y^{(n)} - s^{(n)}\right)^2$$

We want to find an m *and* c such that the above is minimized

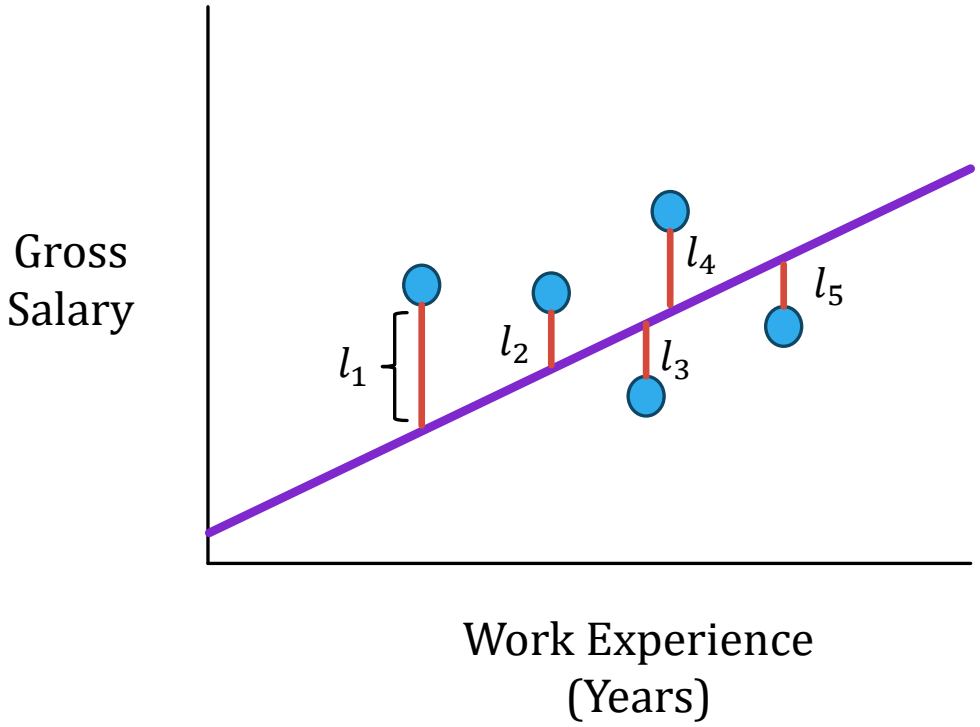
The Objective Function



$$J(m,c) = \frac{1}{2}\sum_{n=1}^{N}\left(mx^{(n)} + c - s^{(n)}\right)^2$$

We want to find an m *and* c such that the above is minimized

The Objective Function

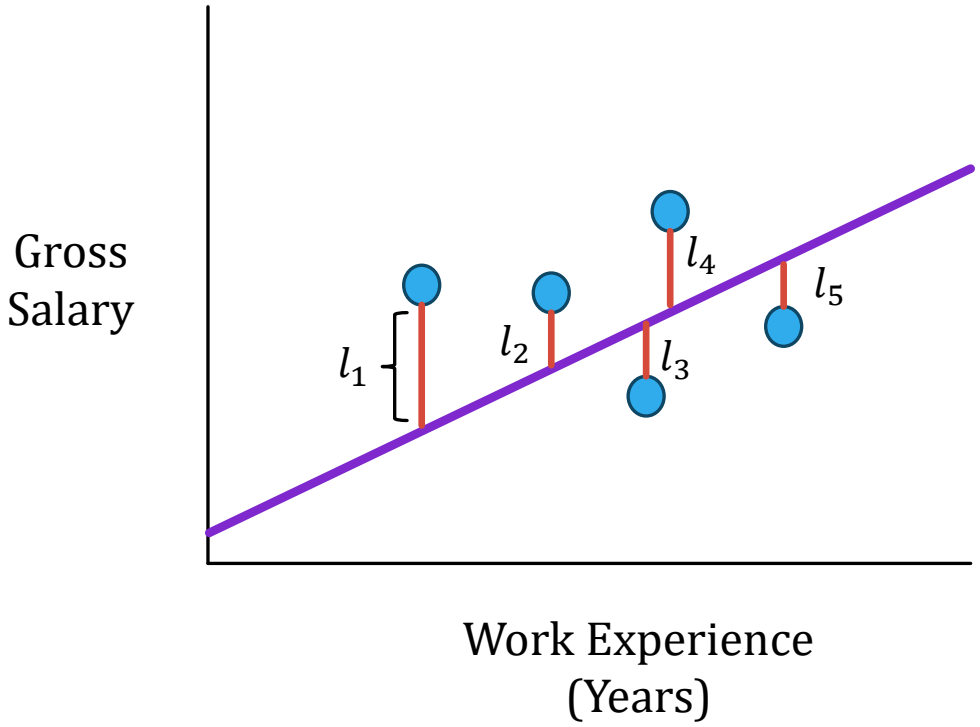


$$J(m,c) = \frac{1}{2}\sum_{n=1}^{N}\left(y^{(n)} - s^{(n)}\right)^2$$

We want to find an m *and* c such that the above is minimized

Since we try to minimize the squared error, this is called **least-squares regression**

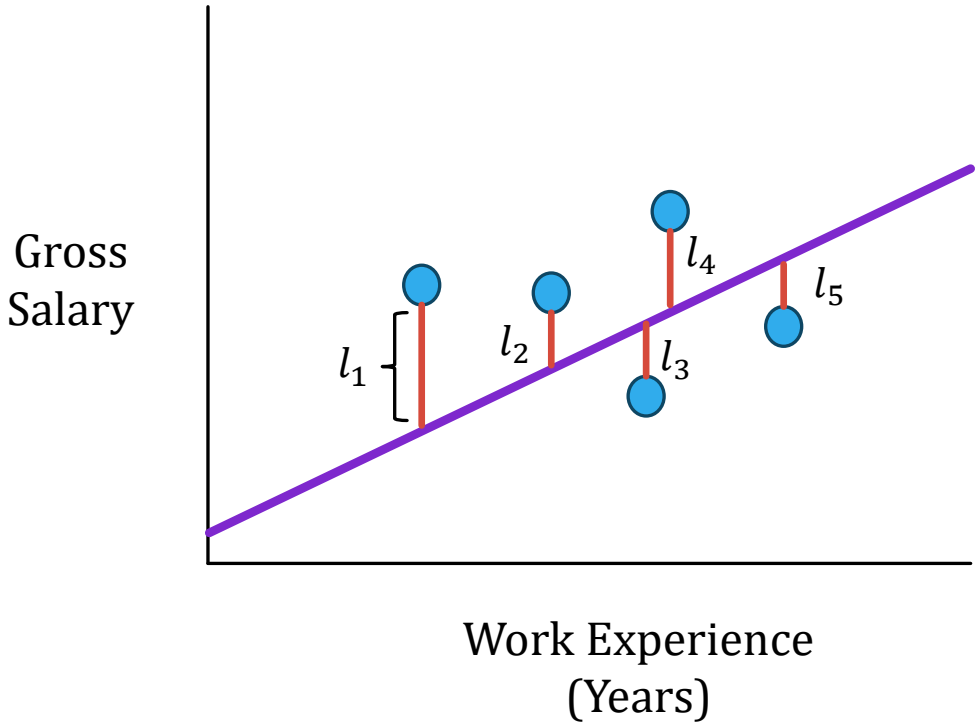
What Do We Want to Learn?



$$J(m,c) = \frac{1}{2}\sum_{n=1}^{N}\left(y^{(n)} - s^{(n)}\right)^2$$

We want to find **(learn)** an m **and** c such that the above is minimized

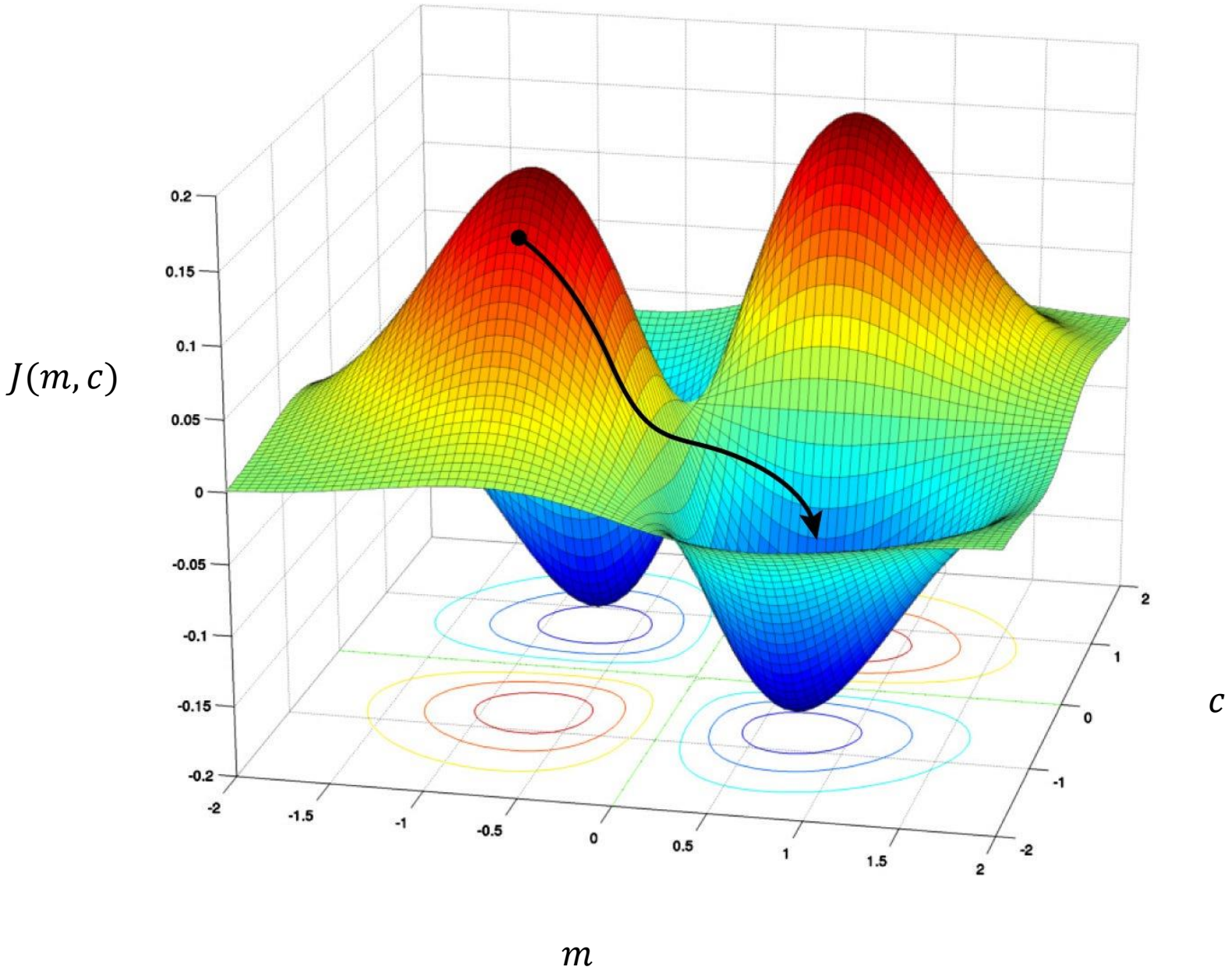
How to Learn?



$$J(m,c) = \frac{1}{2}\sum_{n=1}^{N}\left(mx^{(n)} + c - s^{(n)}\right)^2$$

Let's see

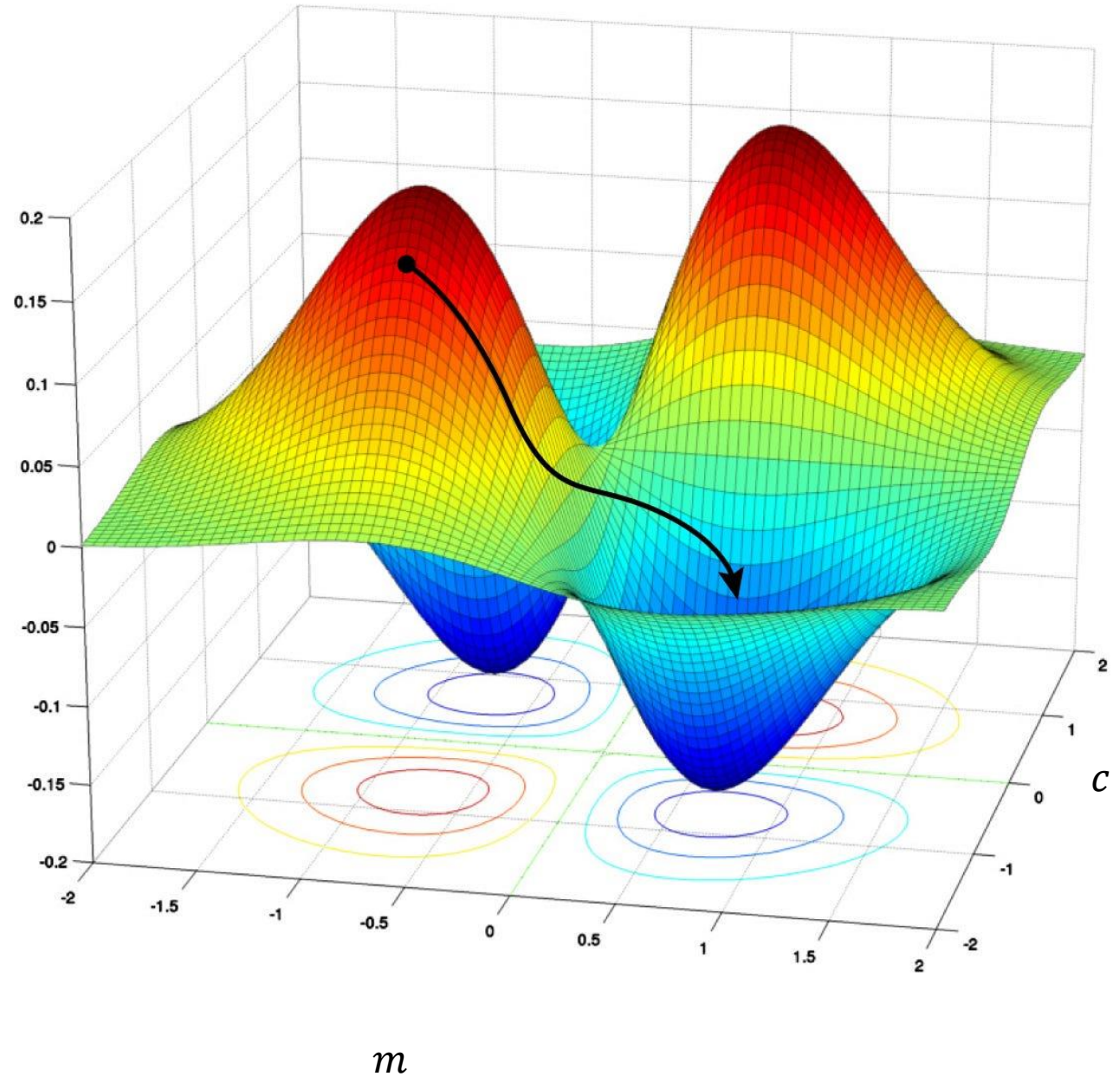
How to Minimize $J(m, c)$



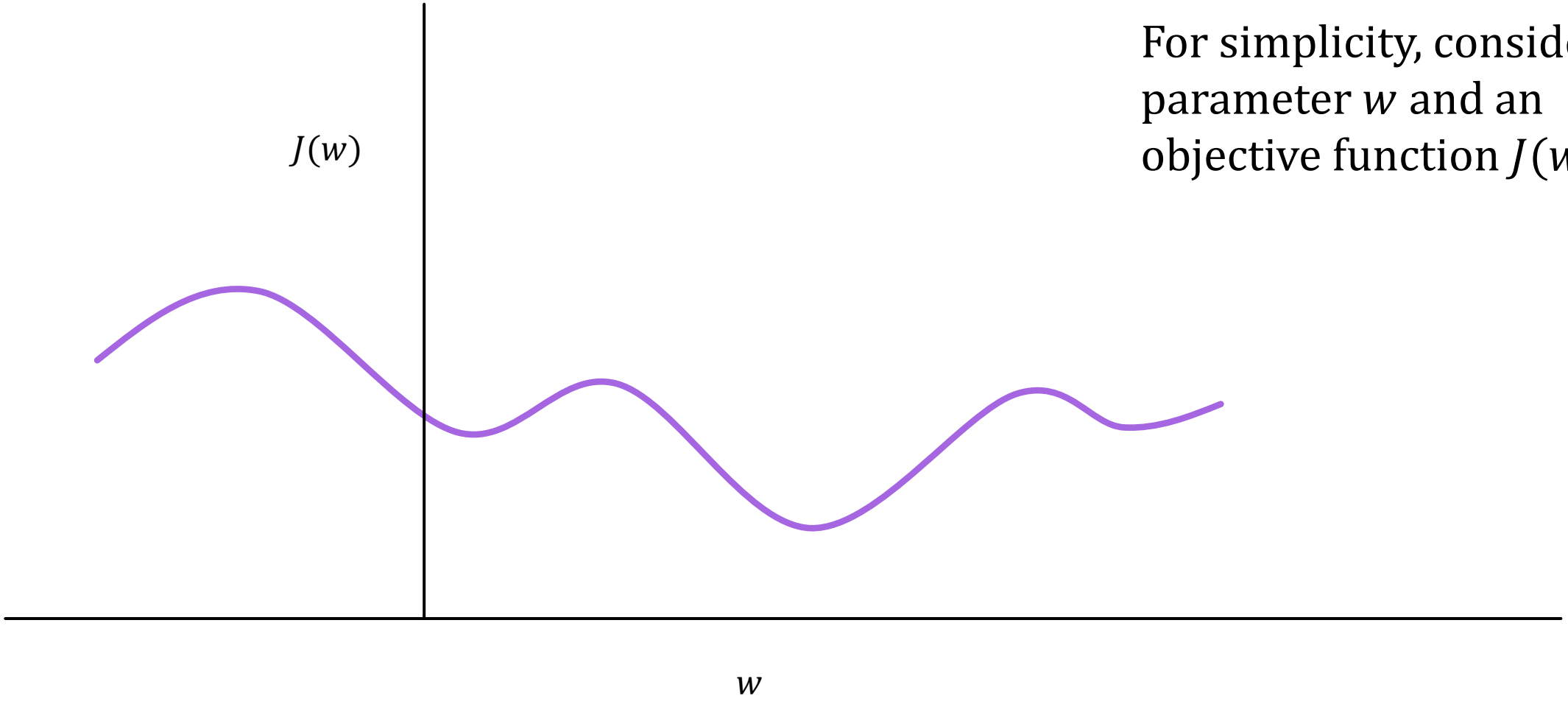
How to Minimize $J(m, c)$

The Goal of Training is to get an m and c (parameters) that would minimize the objective function

Our goal is to reach the global minima

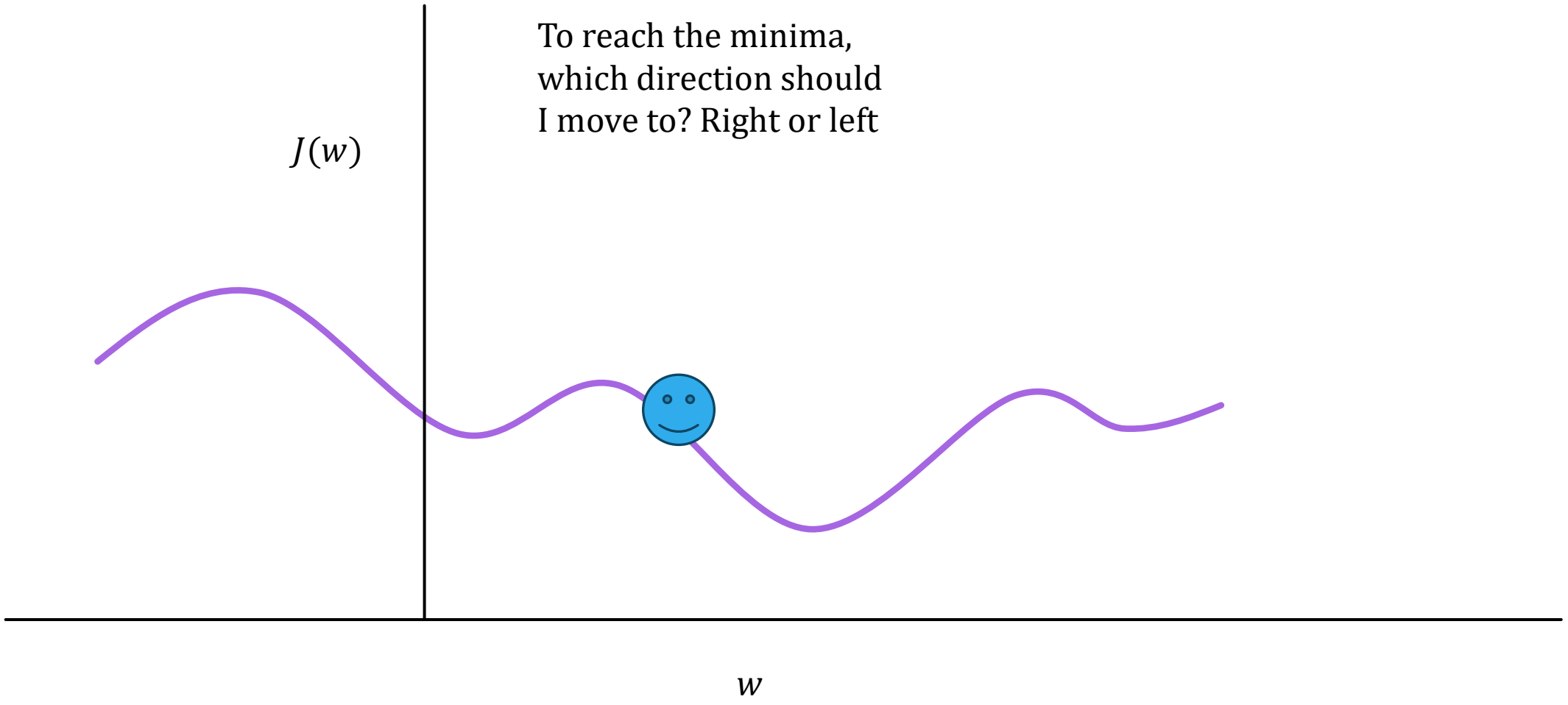


How to Reach the Global Minima: A 2D Example

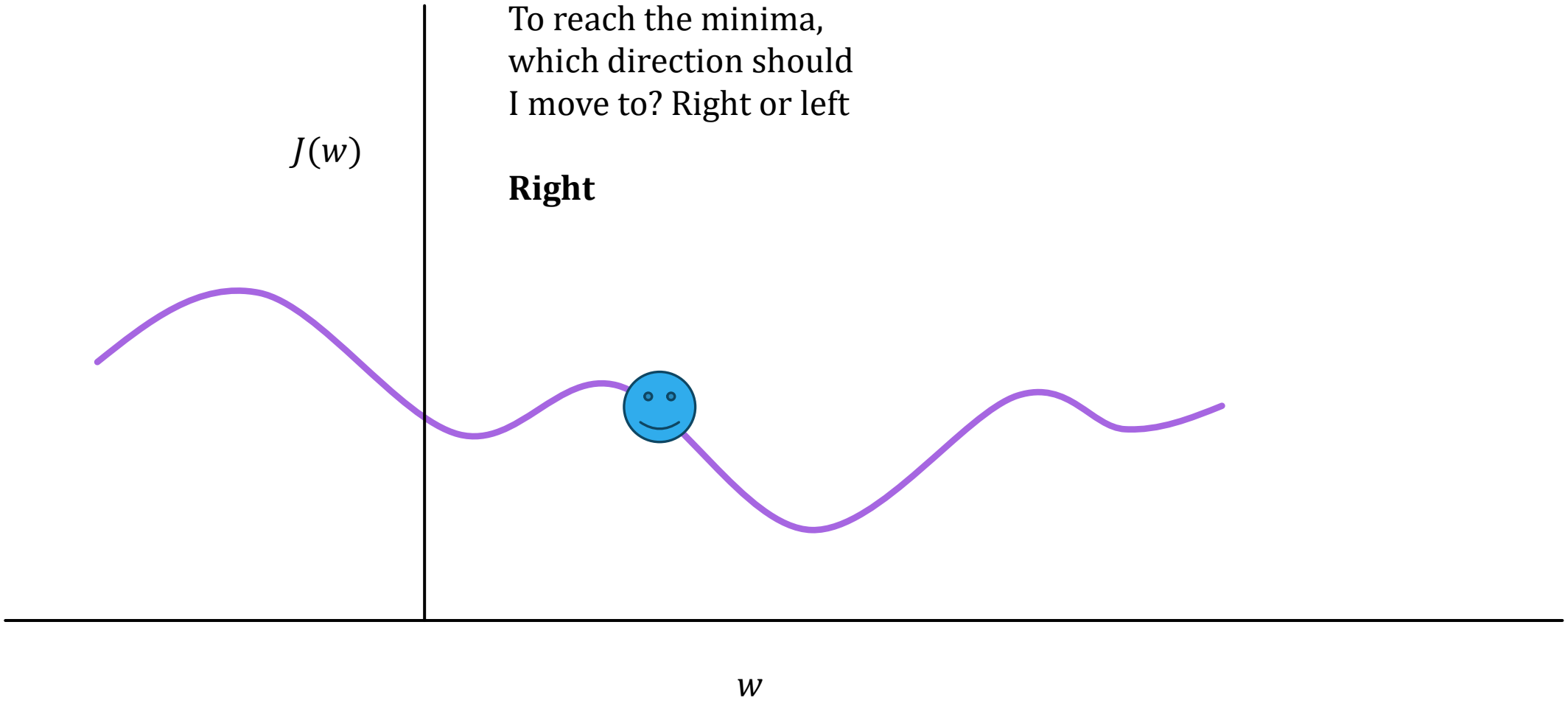


For simplicity, consider a parameter w and an objective function $J(w)$

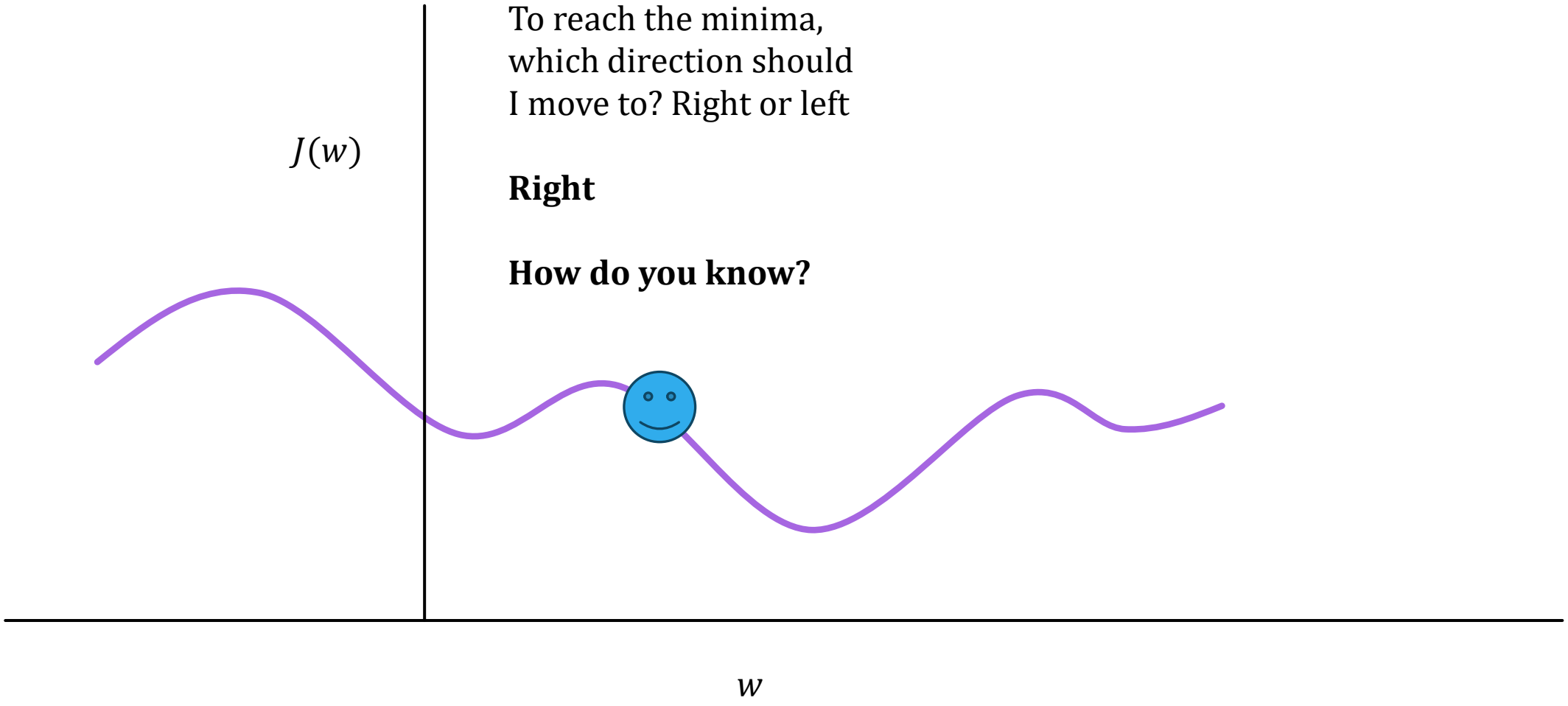
How to Reach the Global Minima: A 2D Example



How to Reach the Global Minima: A 2D Example



How to Reach the Global Minima: A 2D Example

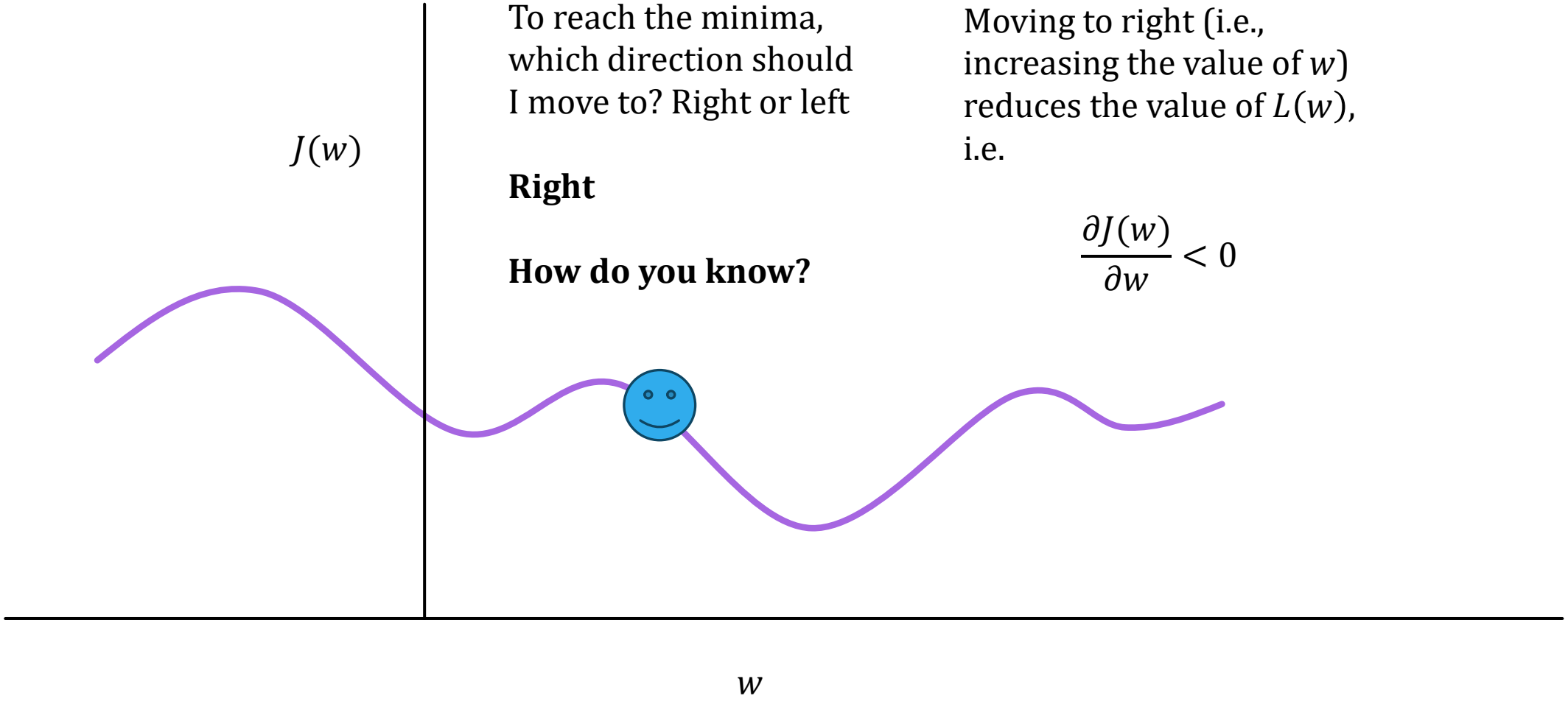


To reach the minima, which direction should I move to? Right or left

Right

How do you know?

How to Reach the Global Minima: A 2D Example



To reach the minima, which direction should I move to? Right or left

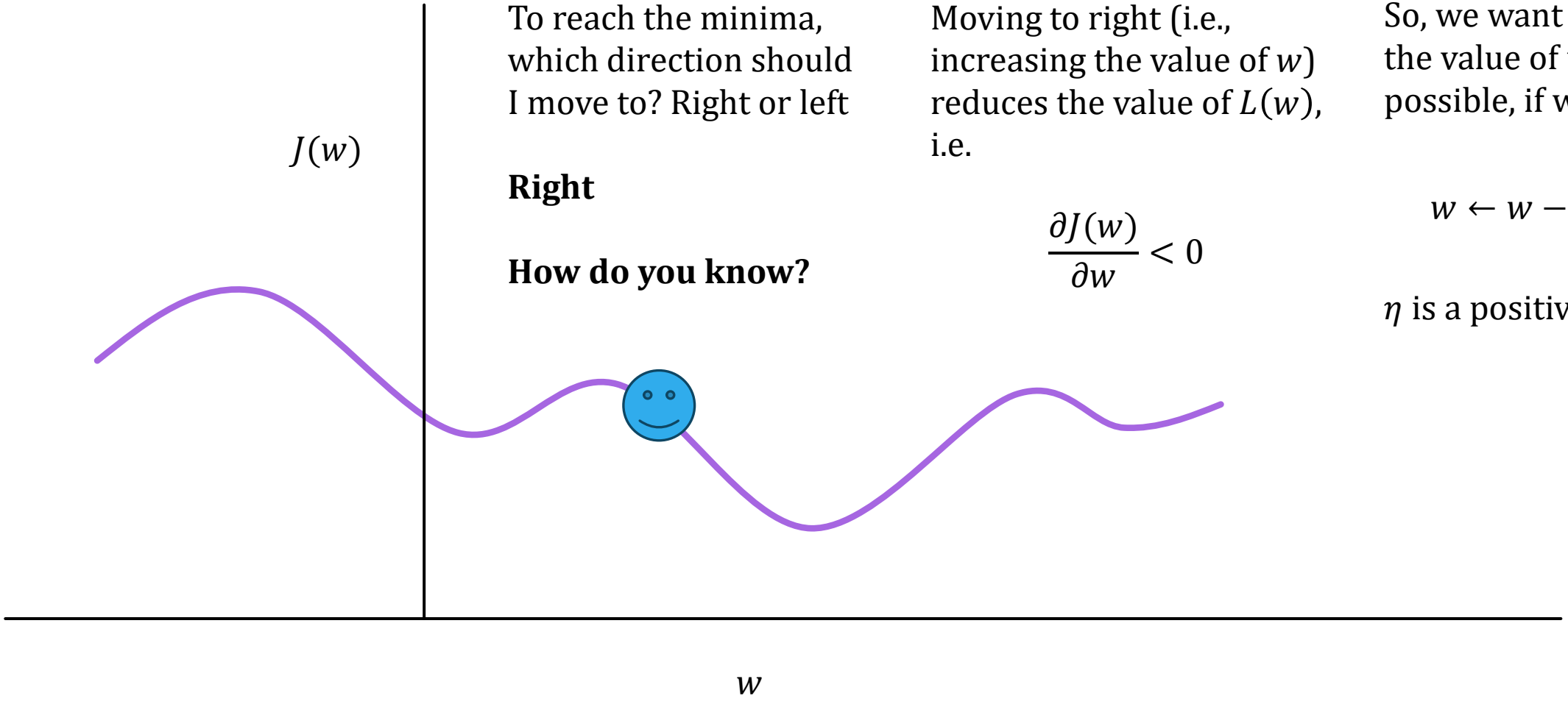
Right

How do you know?

Moving to right (i.e., increasing the value of w) reduces the value of $L(w)$, i.e.

$$\frac{\partial J(w)}{\partial w} < 0$$

How to Reach the Global Minima: A 2D Example



To reach the minima, which direction should I move to? Right or left

Right

How do you know?

Moving to right (i.e., increasing the value of w) reduces the value of $L(w)$, i.e.

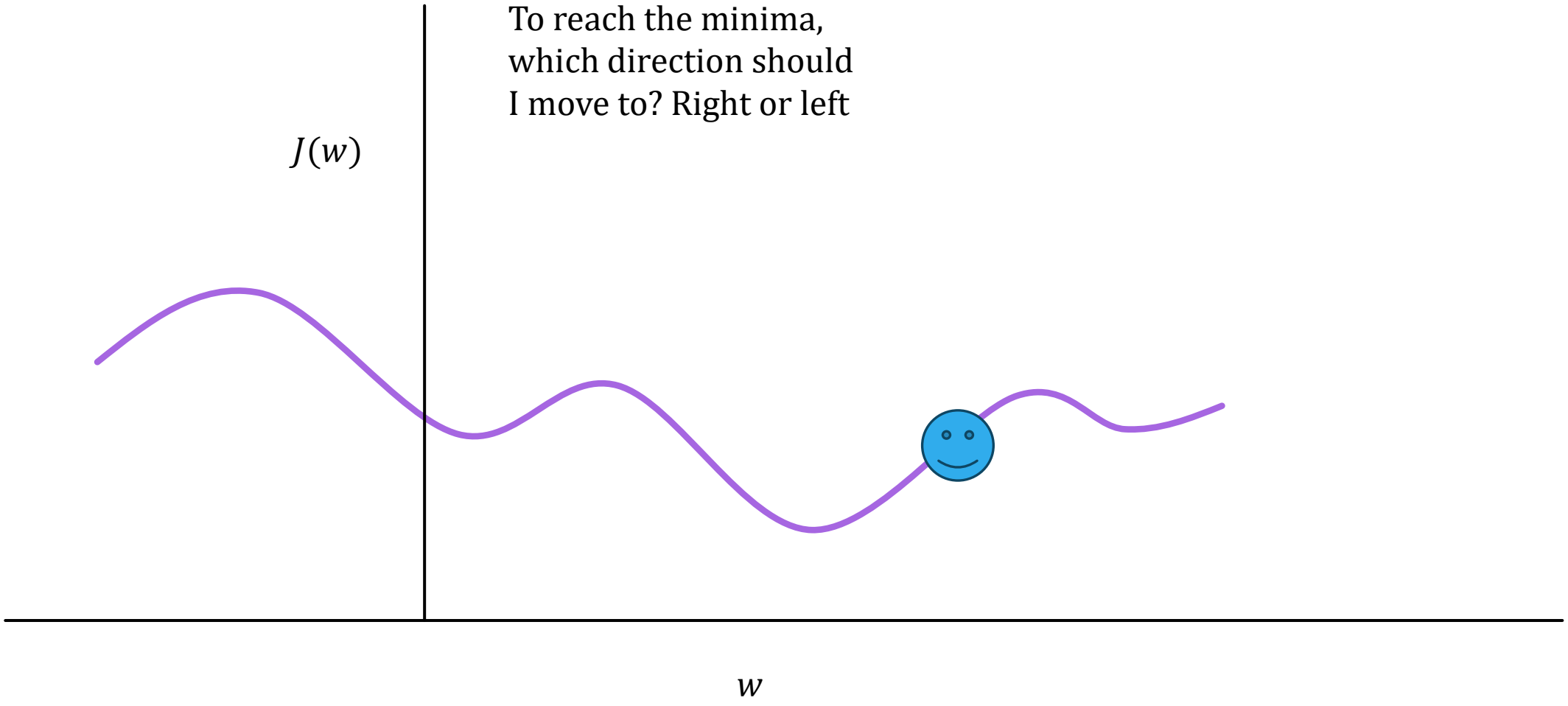
$$\frac{\partial J(w)}{\partial w} < 0$$

So, we want to increase the value of w. That is possible, if we do

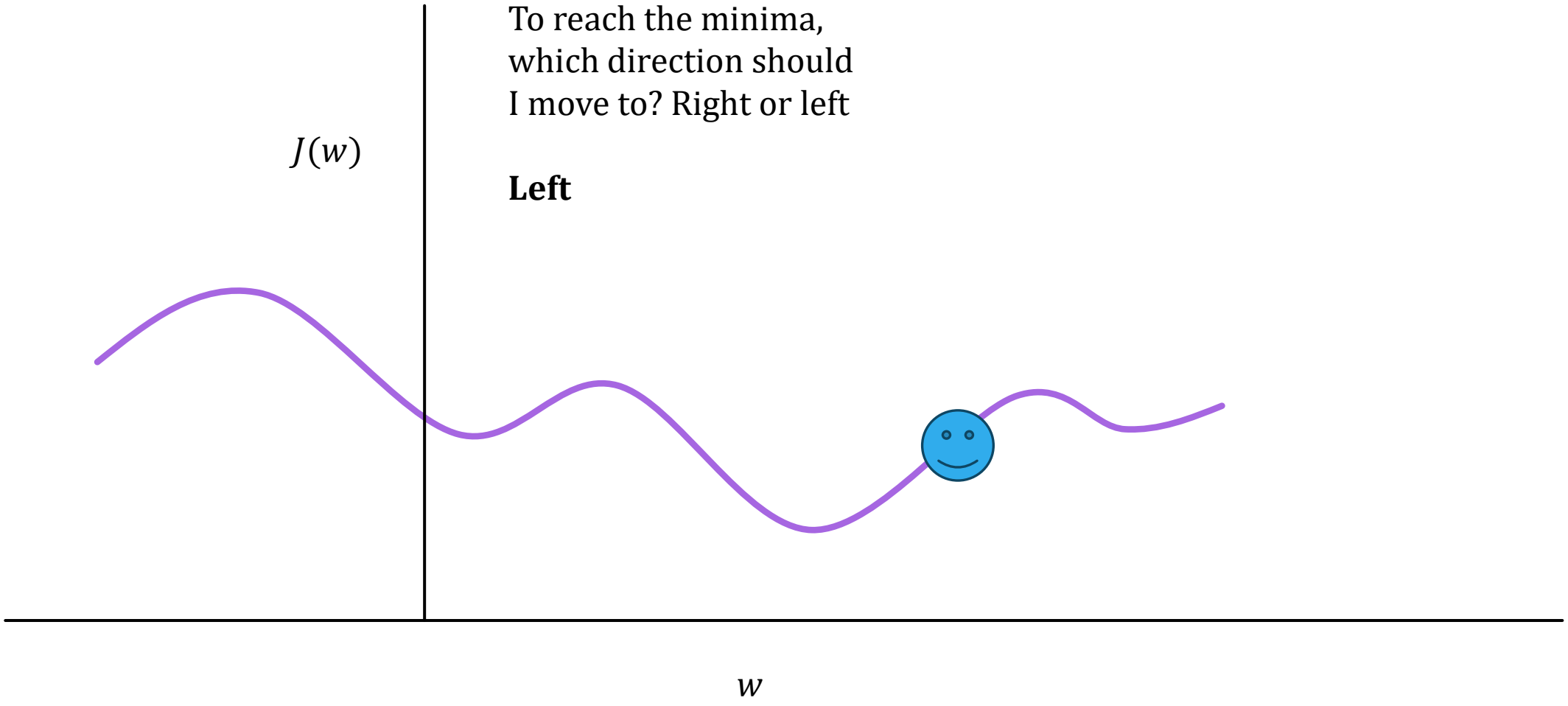
$$w \leftarrow w - \eta \frac{\partial J(w)}{\partial w}$$

η is a positive constant

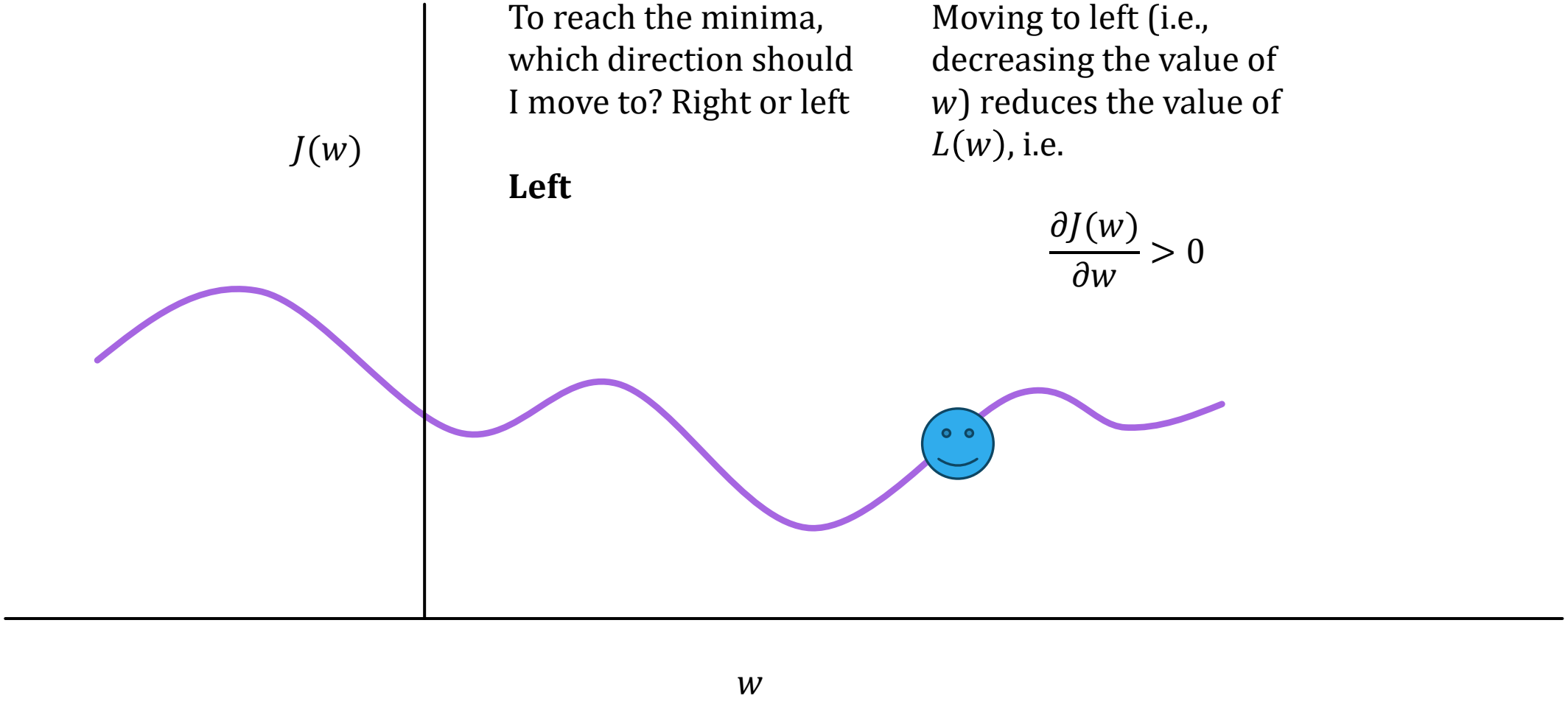
How to Reach the Global Minima: A 2D Example



How to Reach the Global Minima: A 2D Example



How to Reach the Global Minima: A 2D Example



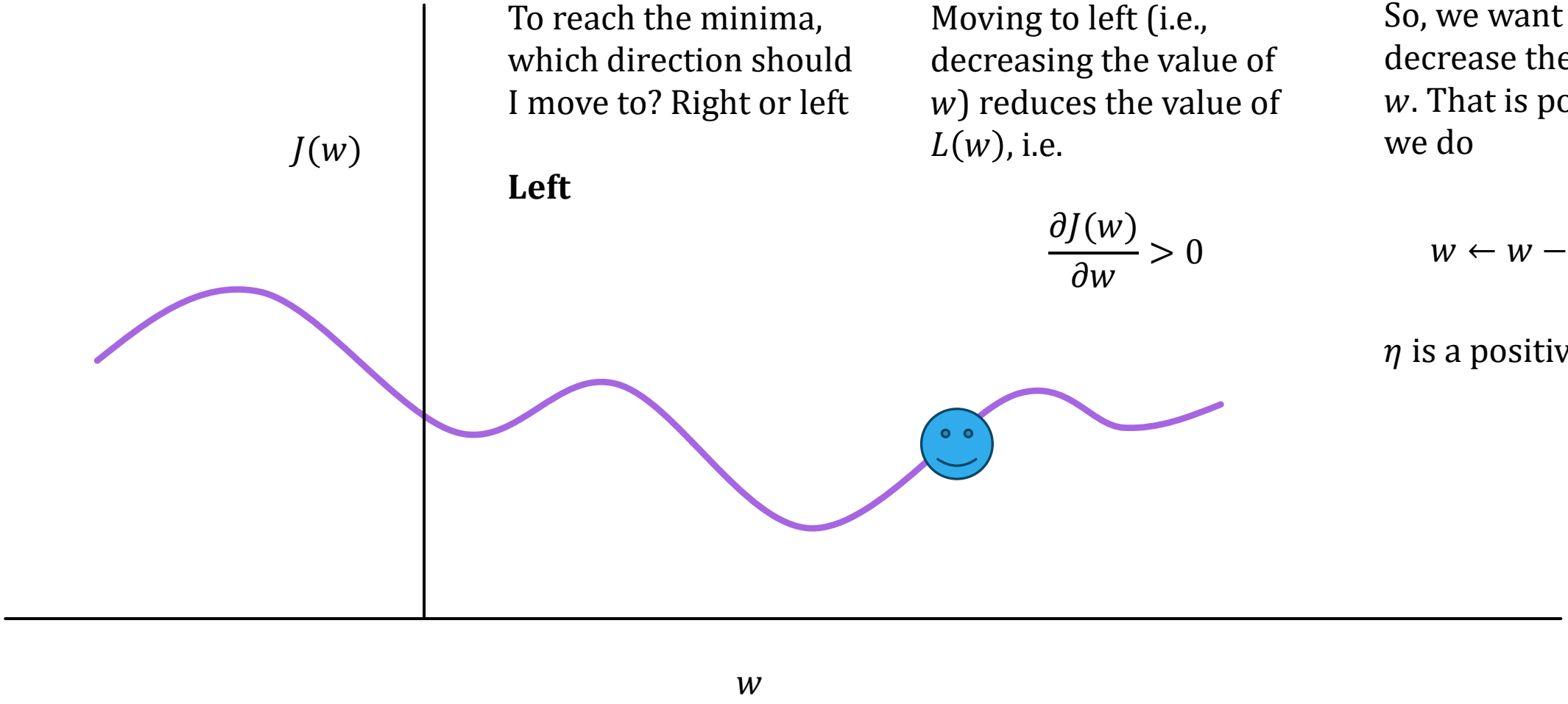
To reach the minima, which direction should I move to? Right or left

Left

Moving to left (i.e., decreasing the value of w) reduces the value of $L(w)$, i.e.

$$\frac{\partial J(w)}{\partial w} > 0$$

How to Reach the Global Minima: A 2D Example



To reach the minima, which direction should I move to? Right or left

Left

Moving to left (i.e., decreasing the value of w) reduces the value of $L(w)$, i.e.

$$\frac{\partial J(w)}{\partial w} > 0$$

So, we want to decrease the value of w. That is possible, if we do

$$w \leftarrow w - \eta \frac{\partial J(w)}{\partial w}$$

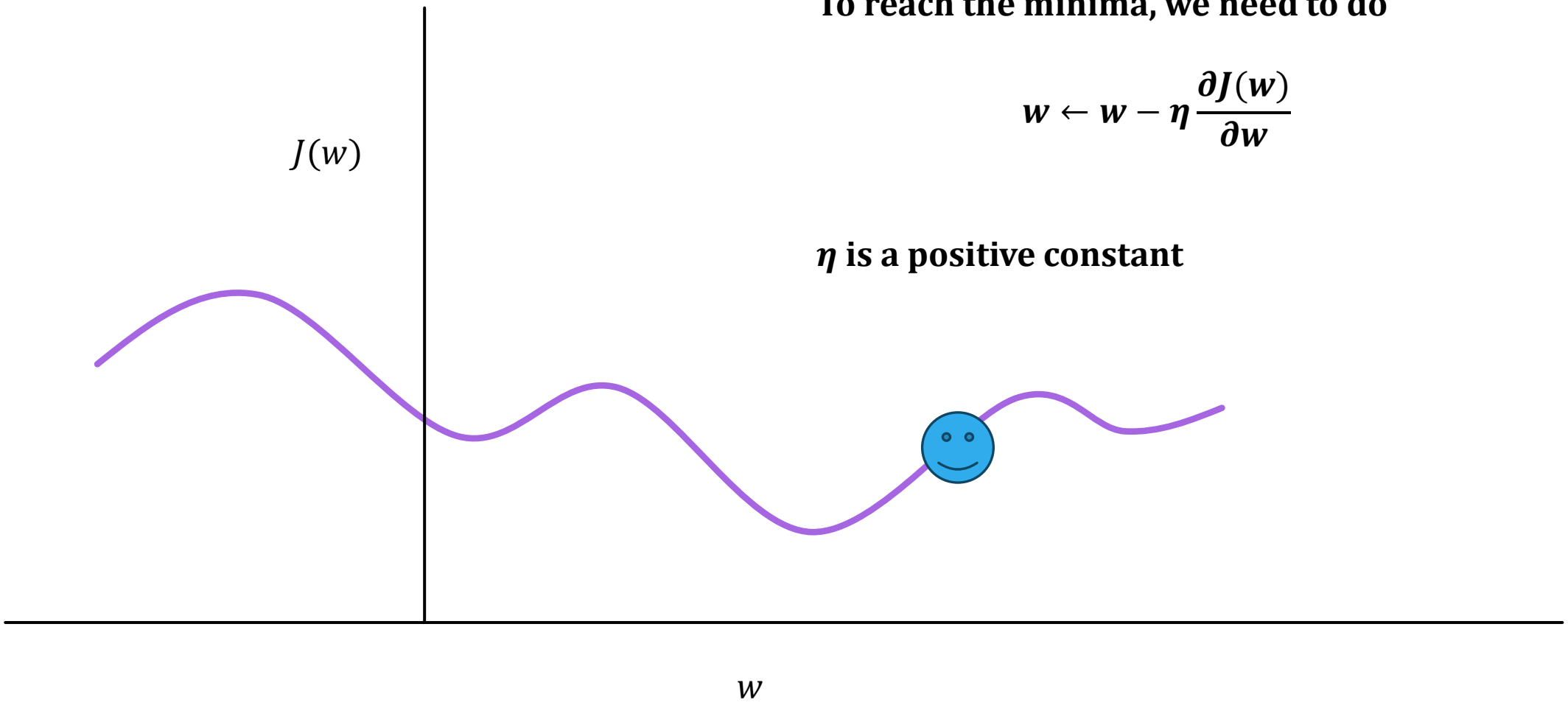
η is a positive constant

How to Reach the Global Minima: A 2D Example

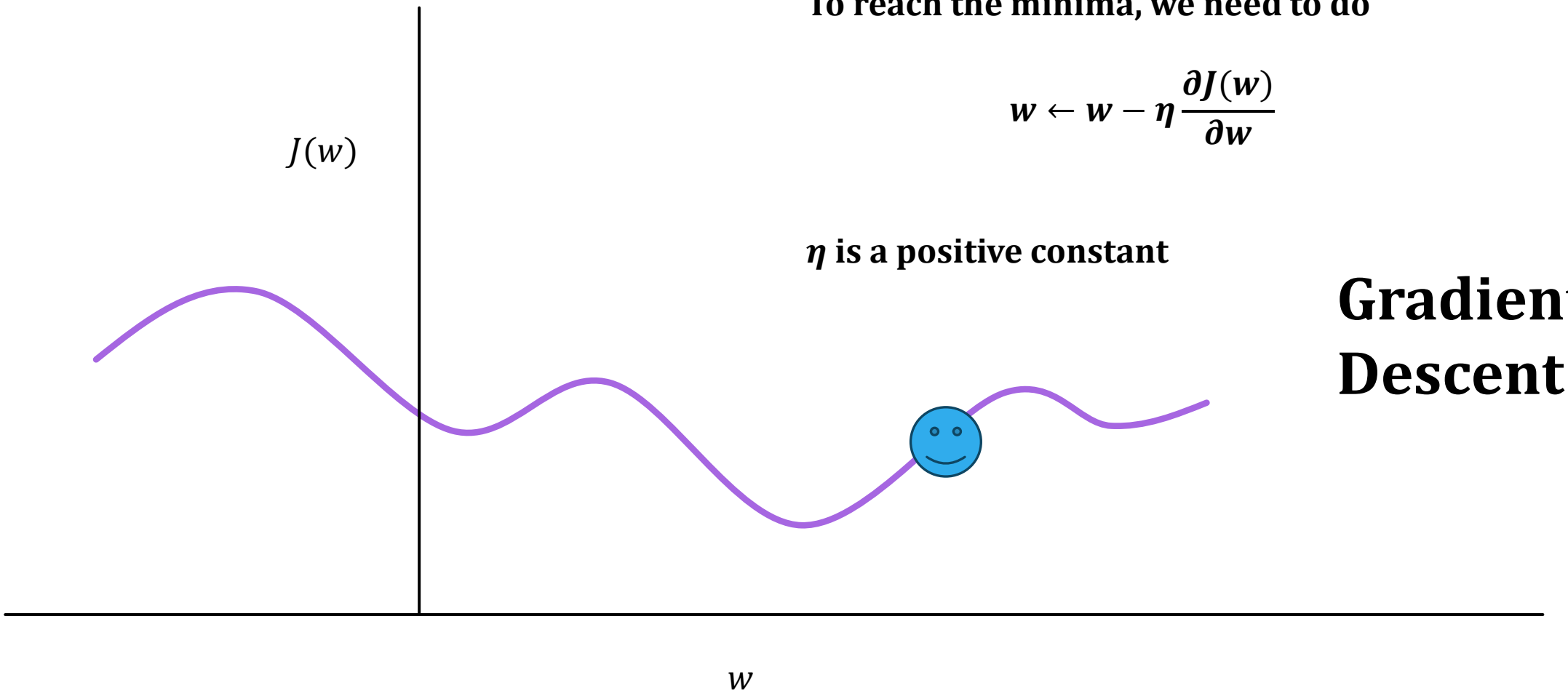
To reach the minima, we need to do

$$\boldsymbol{w \leftarrow w - \eta \frac{\partial J(w)}{\partial w}}$$

η is a positive constant



How to Reach the Global Minima: A 2D Example



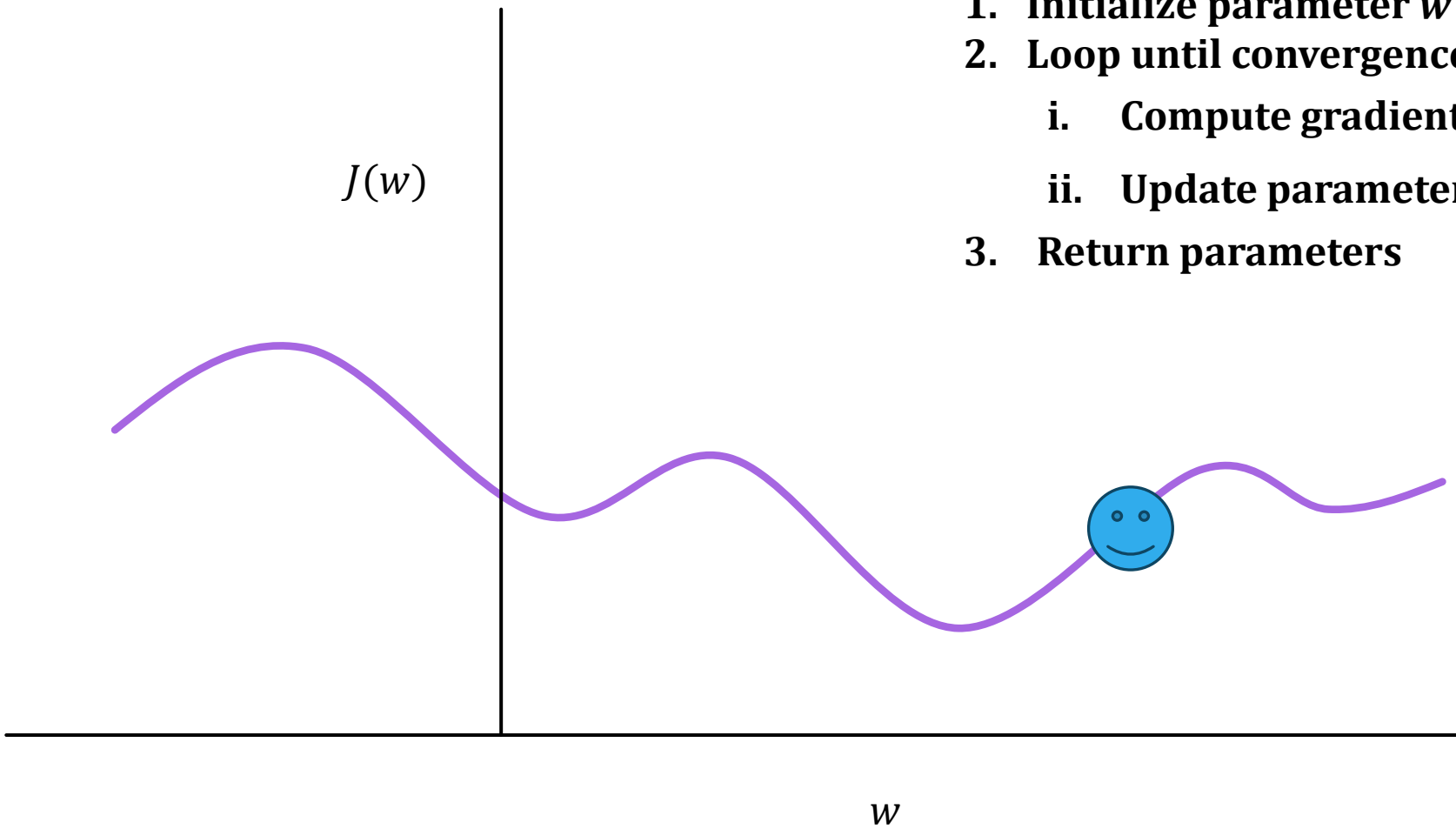
To reach the minima, we need to do

$$w \leftarrow w - \eta \frac{\partial J(w)}{\partial w}$$

η is a positive constant

Gradient Descent

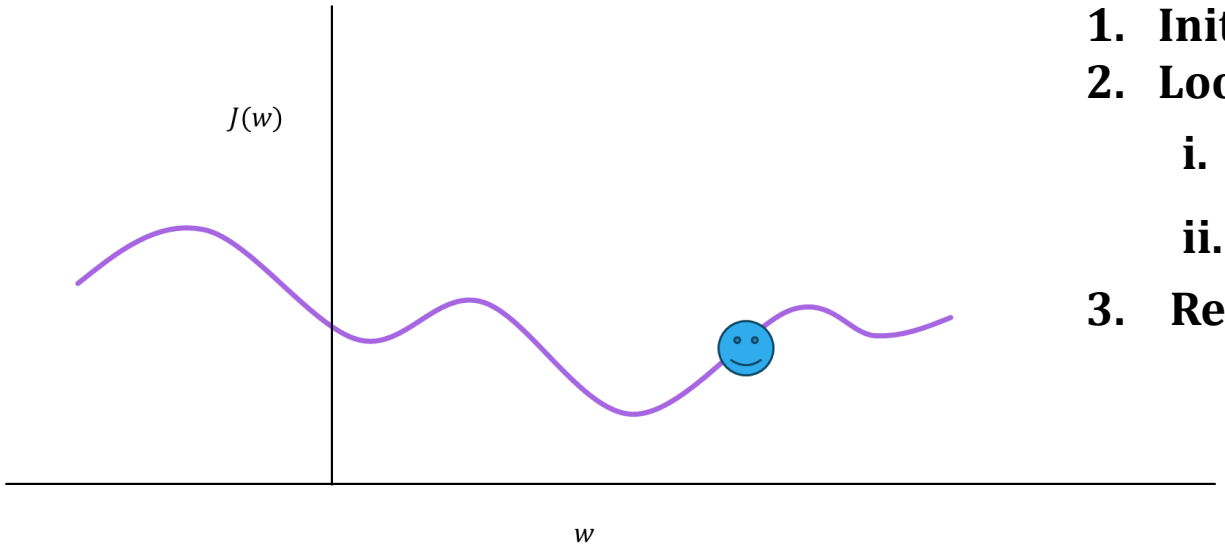
Gradient Descent



1. **Initialize parameter w**
2. **Loop until convergence**
   i. **Compute gradient $\frac{\partial J(w)}{\partial w}$**
   ii. **Update parameters $w \leftarrow w - \eta \frac{\partial J(w)}{\partial w}$**
3. **Return parameters**

η is a hyperparameter whose value should be set using validation performance

Gradient Descent

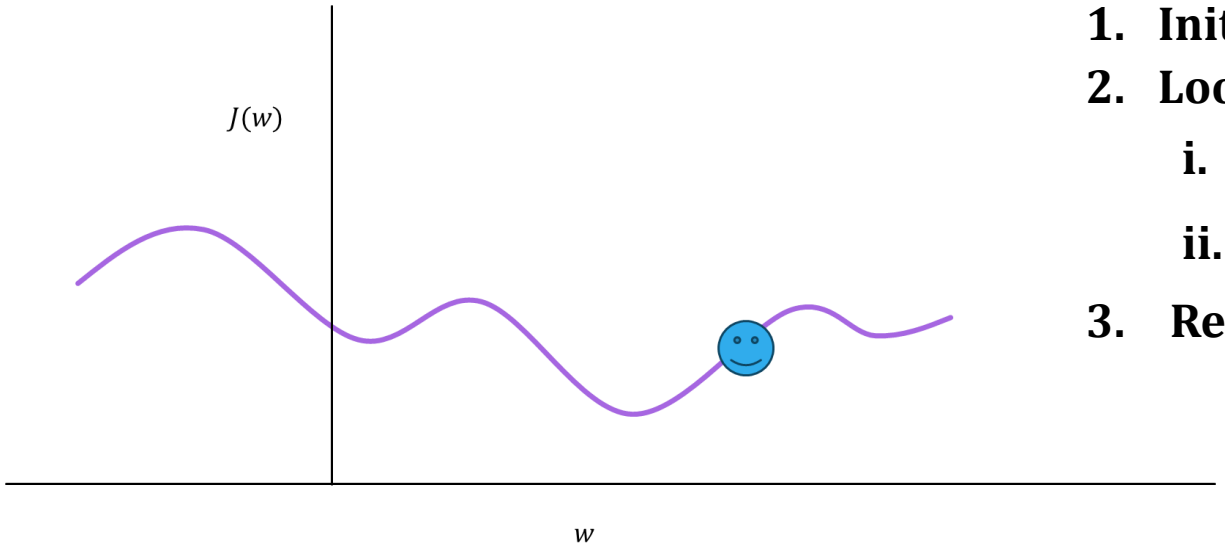


1. **Initialize parameter w**
2. **Loop until convergence**
   i. **Compute gradient $\frac{\partial J(w)}{\partial w}$**
   ii. **Update parameters $w \leftarrow w - \eta \frac{\partial J(w)}{\partial w}$**
3. **Return parameters**

This approach is equally applicable for multiple linear regression

$$y = m_1 x_1 + m_2 x_2 + \cdots + m_p x_p + c$$

Gradient Descent with Many Parameters

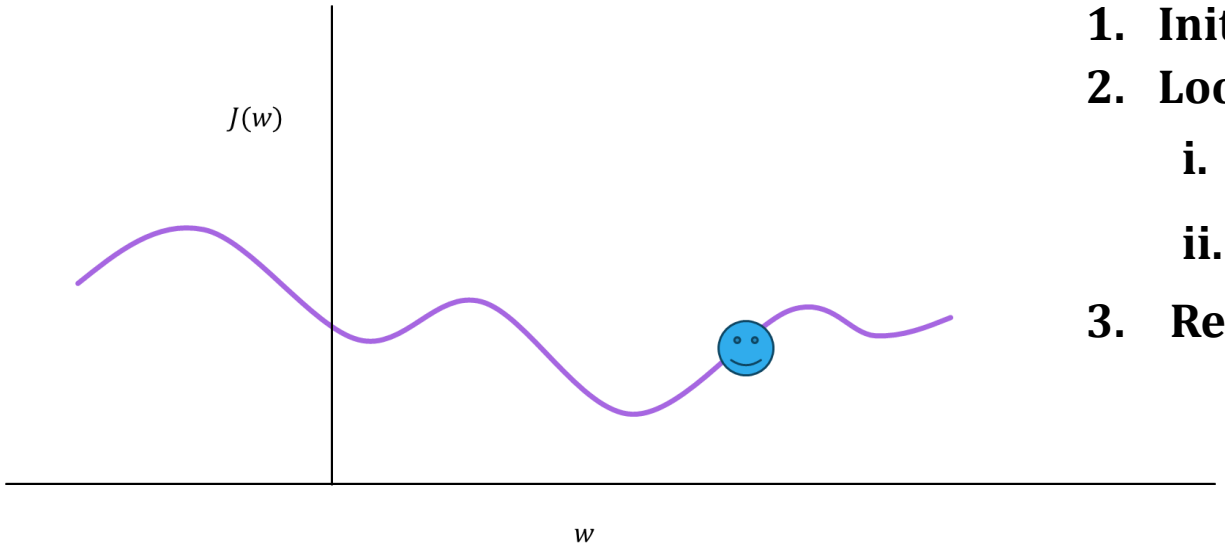


1. **Initialize parameter $m = \{m_1, m_2, \dots, m_p\}$**
2. **Loop until convergence**
   i. **Compute gradient $\frac{\partial J(m)}{\partial m}$**
   ii. **Update parameters $\theta_i \leftarrow \theta_i - \eta \frac{\partial J(m)}{\partial m_i}$**
3. **Return parameters**

Consider the n^{th} data point for multiple linear regression

$$y^{(n)} = m_1 x_1^{(n)} + m_2 x_2^{(n)} + \cdots + m_p x_p^{(n)} + c$$

Gradient Descent with Many Parameters



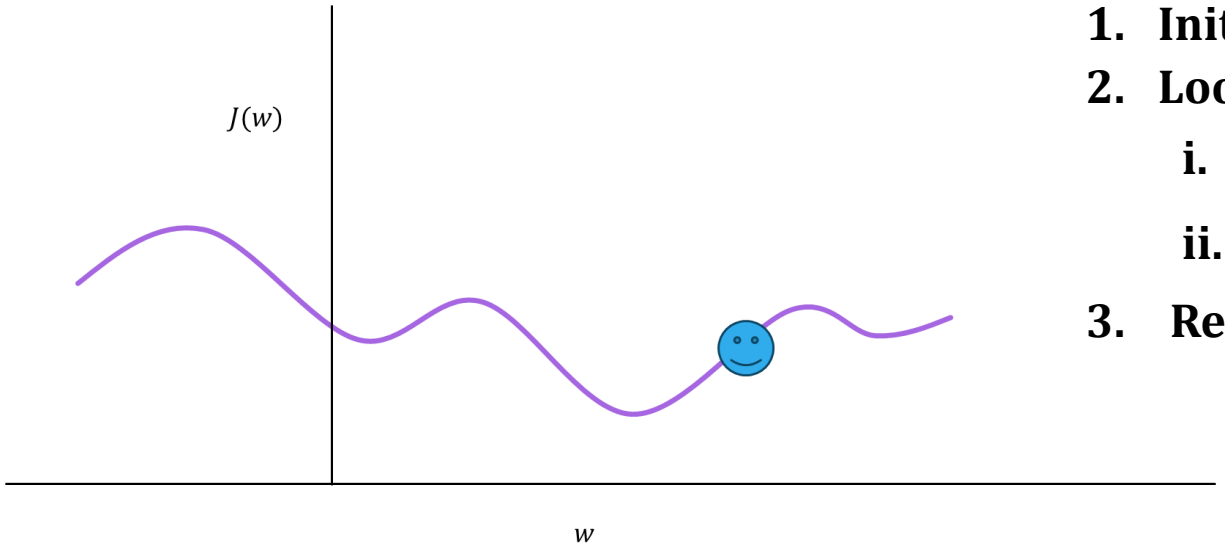
1. **Initialize parameter $m = \{m_1, m_2, \dots, m_p\}$**
2. **Loop until convergence**
   i. **Compute gradient $\frac{\partial J(m)}{\partial m}$**
   ii. **Update parameters $\theta_i \leftarrow \theta_i - \eta \frac{\partial J(m)}{\partial m_i}$**
3. **Return parameters**

$$J(m) = \frac{1}{2} \sum_{n=1}^{N} \left(y^{(n)} - s^{(n)} \right)^2$$

Find the expression for update

$$\theta_i \leftarrow \theta_i - \eta \frac{\partial J(m)}{\partial m_i}$$

Gradient Descent with Many Parameters



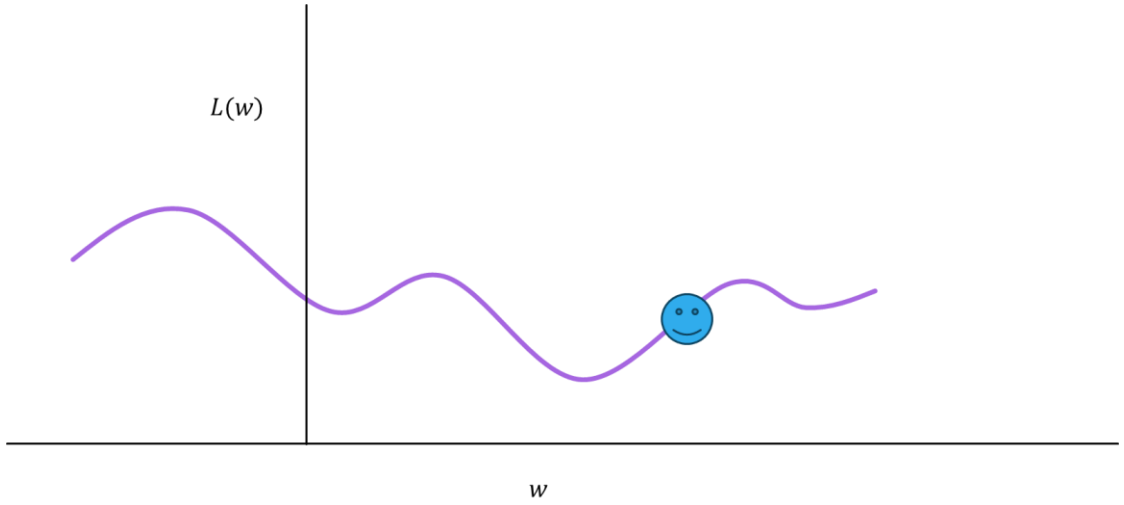
1. **Initialize parameter $m = \{m_1, m_2, \dots, m_p\}$**
2. **Loop until convergence**
   i. **Compute gradient $\frac{\partial J(m)}{\partial m}$**
   ii. **Update parameters $\theta_i \leftarrow \theta_i - \eta \frac{\partial J(m)}{\partial m_i}$**
3. **Return parameters**

$$J(m) = \frac{1}{2}\sum_{n=1}^{N}\left(y^{(n)} - s^{(n)}\right)^2$$

Find the expression for update

$$\theta_i \leftarrow \theta_i - \eta \sum_{n=1}^{N}\left(y^{(n)} - s^{(n)}\right)x_i^{(n)}$$

Gradient Descent: Challenges

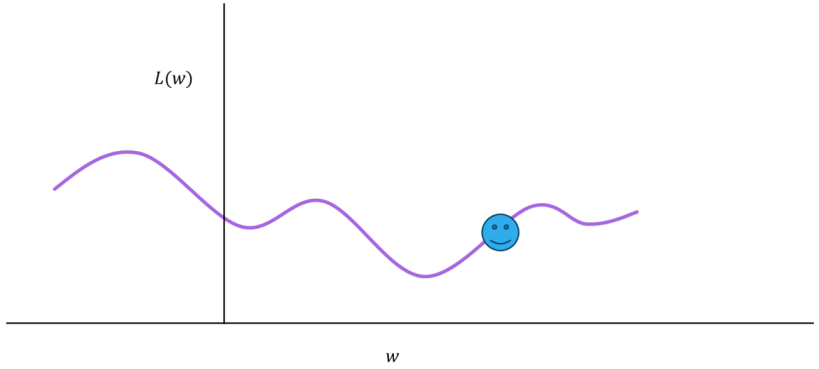


1. **Initialize parameters**
2. **Loop until convergence**
   i. **Calculate loss $J(w)$**
   ii. **Compute gradient $\frac{\partial J(w)}{\partial w}$**
   iii. **Update parameters $w \leftarrow w - \eta \frac{\partial J(w)}{\partial w}$**
3. **Return parameters**

In Batch Gradient Descent with N training samples (data points), we compute the total objective function considering all the N training samples

$$J(m, c) = \frac{1}{2} \sum_{n=1}^{N} (mx_n + c - s_n)^2$$

Gradient Descent: Challenges



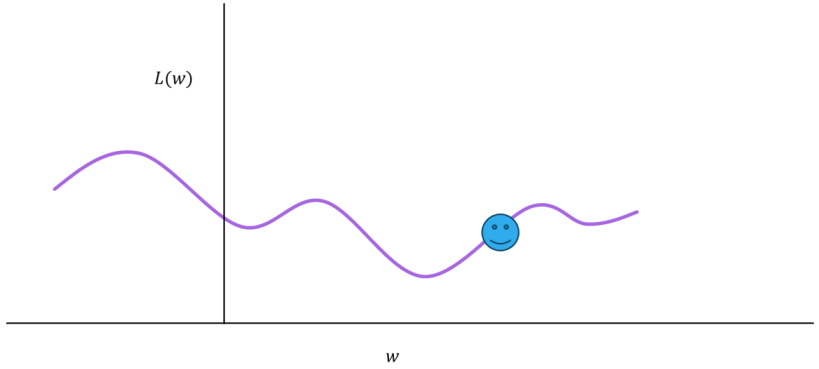
1. **Initialize parameters**
2. **Loop until convergence**
   i. **Calculate loss $J(w)$**
   ii. **Compute gradient $\frac{\partial J(w)}{\partial w}$**
   iii. **Update parameters $w \leftarrow w - \eta \frac{\partial J(w)}{\partial w}$**
3. **Return parameters**

In Batch Gradient Descent with N training samples (data points),

$$J(m, c) = \frac{1}{2} \sum_{n=1}^{N} (mx_n + c - s_n)^2$$

$$\frac{\partial J(w)}{\partial w} = \frac{1}{2} \sum_{n=1}^{N} \frac{\partial}{\partial w} (mx_n + c - s_n)^2$$

Gradient Descent: Challenges



1. **Initialize parameters**
2. **Loop until convergence**
   i. **Calculate loss $J(w)$**
   ii. **Compute gradient $\frac{\partial J(w)}{\partial w}$**
   iii. **Update parameters $w \leftarrow w - \eta \frac{\partial J(w)}{\partial w}$**
3. **Return parameters**

$$J(m,c) = \frac{1}{2}\sum_{n=1}^{N}(mx_n + c - s_n)^2$$

$$\frac{\partial J(w)}{\partial w} = \frac{1}{2}\sum_{n=1}^{N}\frac{\partial}{\partial w}(mx_n + c - s_n)^2$$

For each update, it will take a long time if we have many training samples

We will also have to compute and store N gradients for each update

Solution: Stochastic Gradient Descent

In Batch Gradient Descent with N samples (data points),

$$\boldsymbol{J(m,c) = \sum_{n=1}^{N} \frac{1}{2}(mx_n + c - s_n)^2 = \sum_{n=1}^{N} J_n(m,c)}$$

Suppose, I calculate the objective function for one sample, say and do the parameter update

Do it again for another sample, and continue doing it for individual samples

In this situation, each update would require only one gradient computation (less time, less storage for gradient values)

Stochastic Gradient Descent

1. **Initialize parameters**
2. **Loop until convergence**
   i. **Randomly shuffle samples in the training dataset**
   ii. **For training sample $n = 1, 2, 3, \dots, N$**
      a. **Compute gradient $\frac{\partial J_n(w)}{\partial w}$**
      b. **Update parameters $w \leftarrow w - \eta \frac{\partial J_n(w)}{\partial w}$**
3. **Return parameters**

Challenges in Stochastic and Batch Gradient Descent

- BGD: not data efficient when data is very similar
- BGD: High computational time for each step

- SGD: not computationally efficient
- SGD: noisy results since at a time, one data point is considered

- Solution: minibatch GD

Minibatch Gradient Descent

1. **Initialize parameters**
2. **Loop until convergence**
   i. **Randomly shuffle samples in the training dataset**
   ii. **From training dataset, create b minibatches of size u**
   iii. **For each minibatches $n = 1, 2, 3, \dots, b$**
      a. **Compute gradient $\frac{\partial J_n(w)}{\partial w}$**
      b. **Update parameters $w \leftarrow w - \eta \frac{\partial J_n(w)}{\partial w}$**
3. **Return parameters**

How to Initialize?

1. **Initialize parameters**
2. Loop until convergence
   i. Randomly shuffle samples in the training dataset
   ii. From training dataset, create b minibatches of size u
   iii. For each minibatches $n = 1, 2, 3, \dots, b$
      a. Compute gradient $\frac{\partial J_n(w)}{\partial w}$
      b. Update parameters $w \leftarrow w - \eta \frac{\partial J_n(w)}{\partial w}$
3. Return parameters

Can be all zeros

Can be random