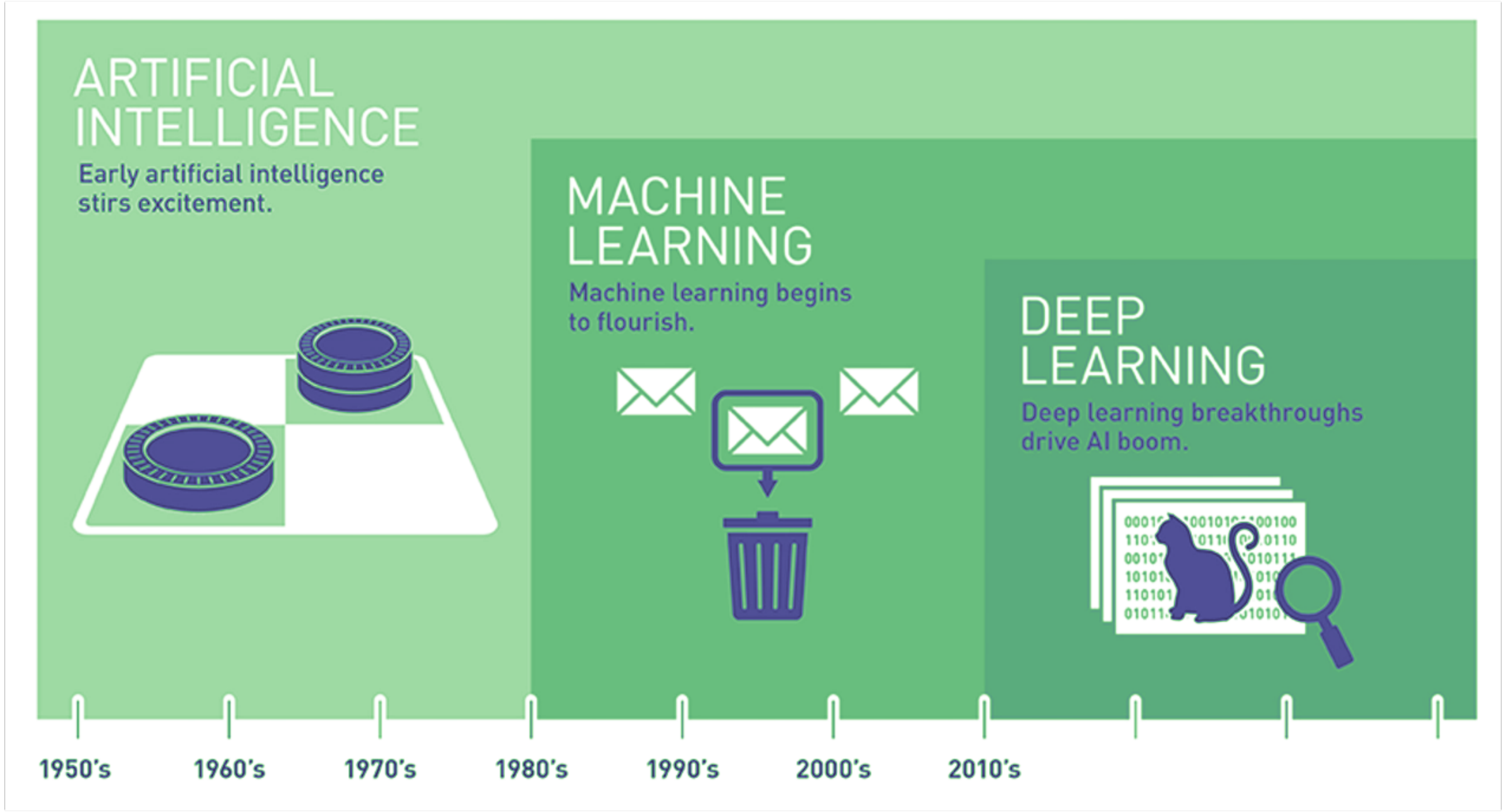
Lecture *1* – Machine leanring basics

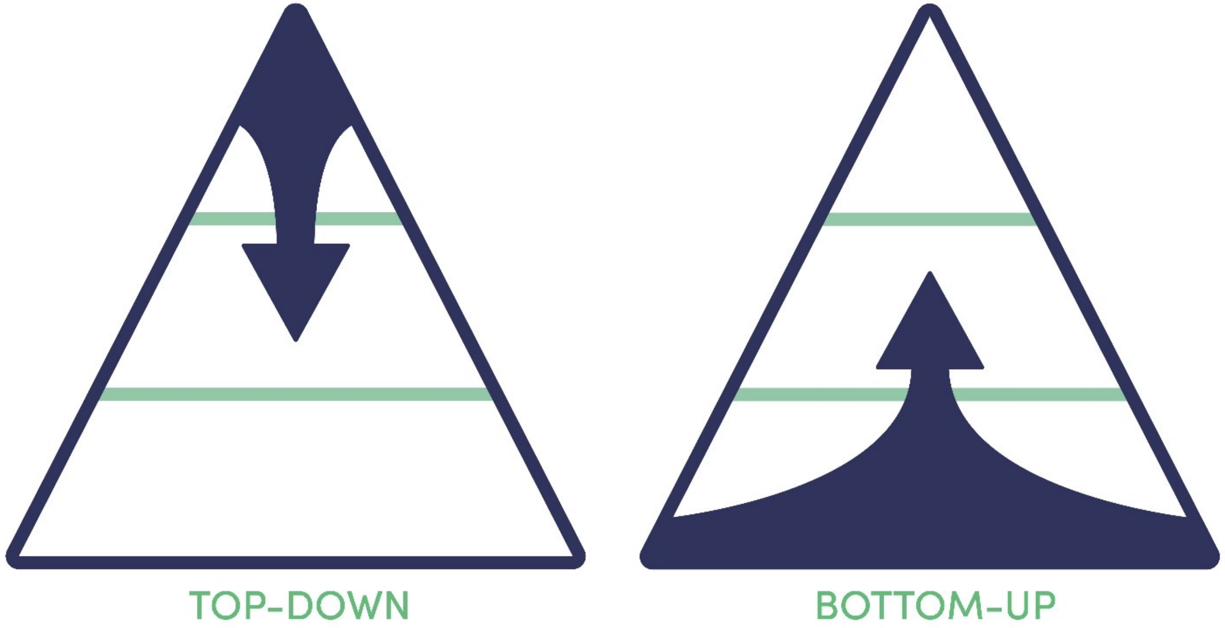
AI in Genetics

ZOO6927 / BOT6935 / ZOO4926

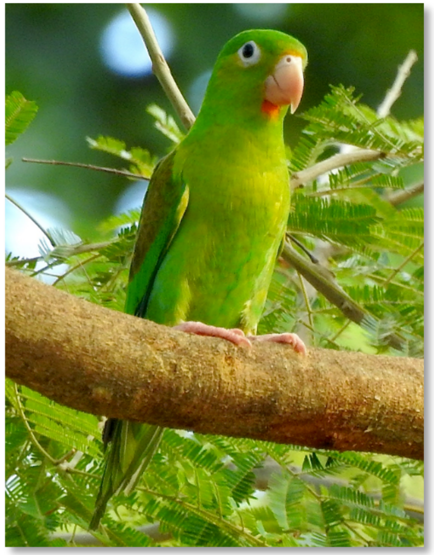
The AI landscape



Source: Copeland, M. Retrieve from https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/


TOP-DOWN
BOTTOM-UP


TOP-DOWN



Laura Gaudette (CC-BY)

Can the orange-chinned parakeet fly?

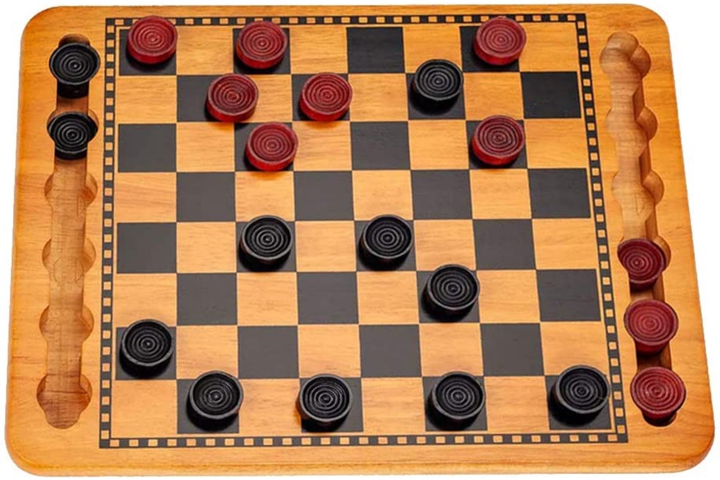
Knowledge base

- Birds can fly (0.99).
- A parrot is a bird.
- The orange-chinned parakeet is a parrot.
- …

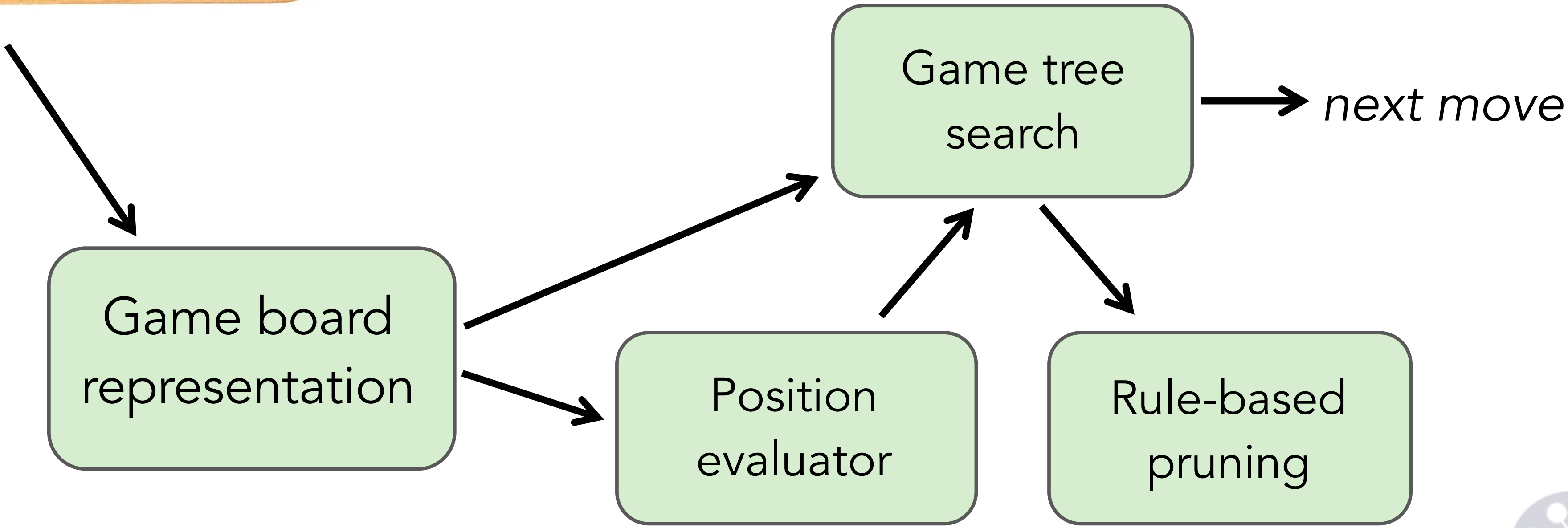
Inference Engine

Q: Can the orange-chinned parakeet fly?

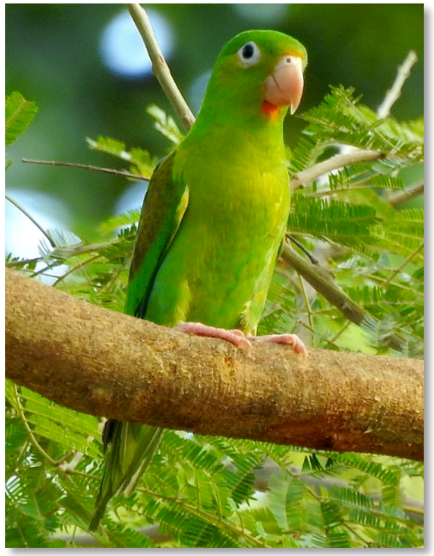
A: The orange-chinned parakeet most likely can fly.

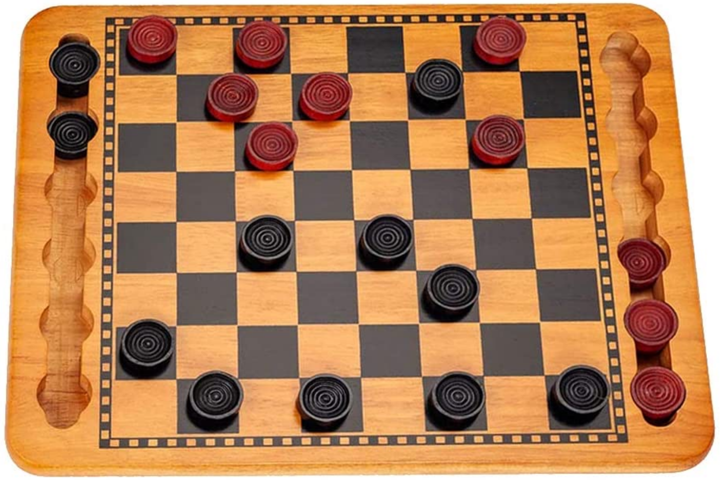


What is red's next move?



Top-Down (Summary)

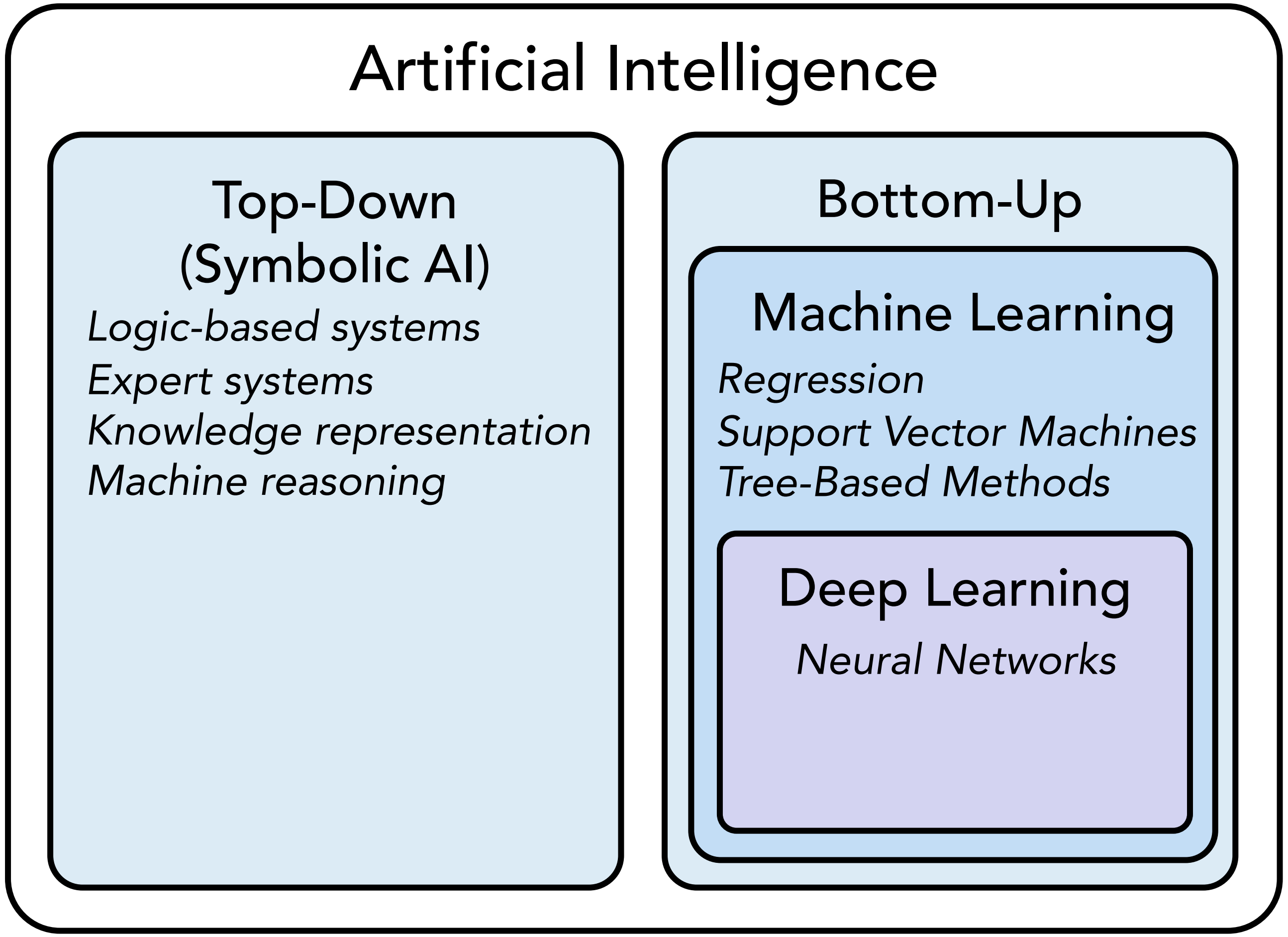




BOTTOM-UP

Cat or Dog?

→ Mathematical function mapping image to label ("cat" or "dog") → *cat*


Artificial Intelligence
Top-Down
(Symbolic AI)
Logic-based systems
Expert systems
Knowledge representation
Machine reasoning
Bottom-Up
Machine Learning
Regression
Support Vector Machines
Tree-Based Methods
Deep Learning
Neural Networks

The Machine Learning landscape

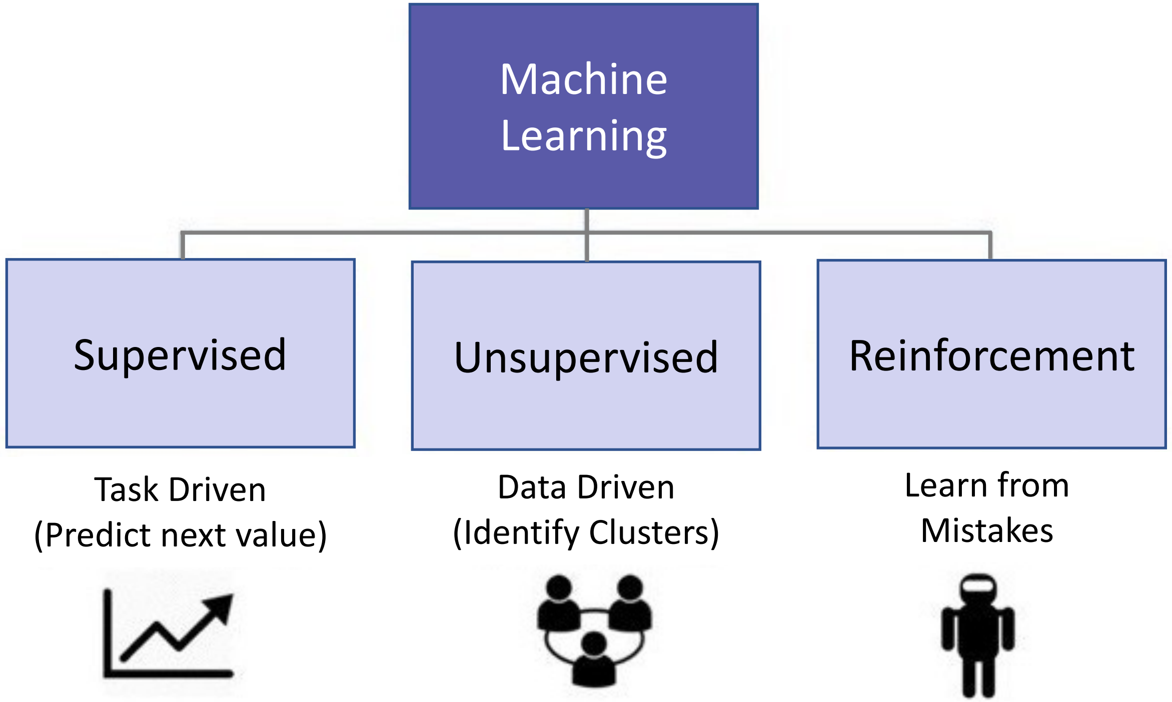
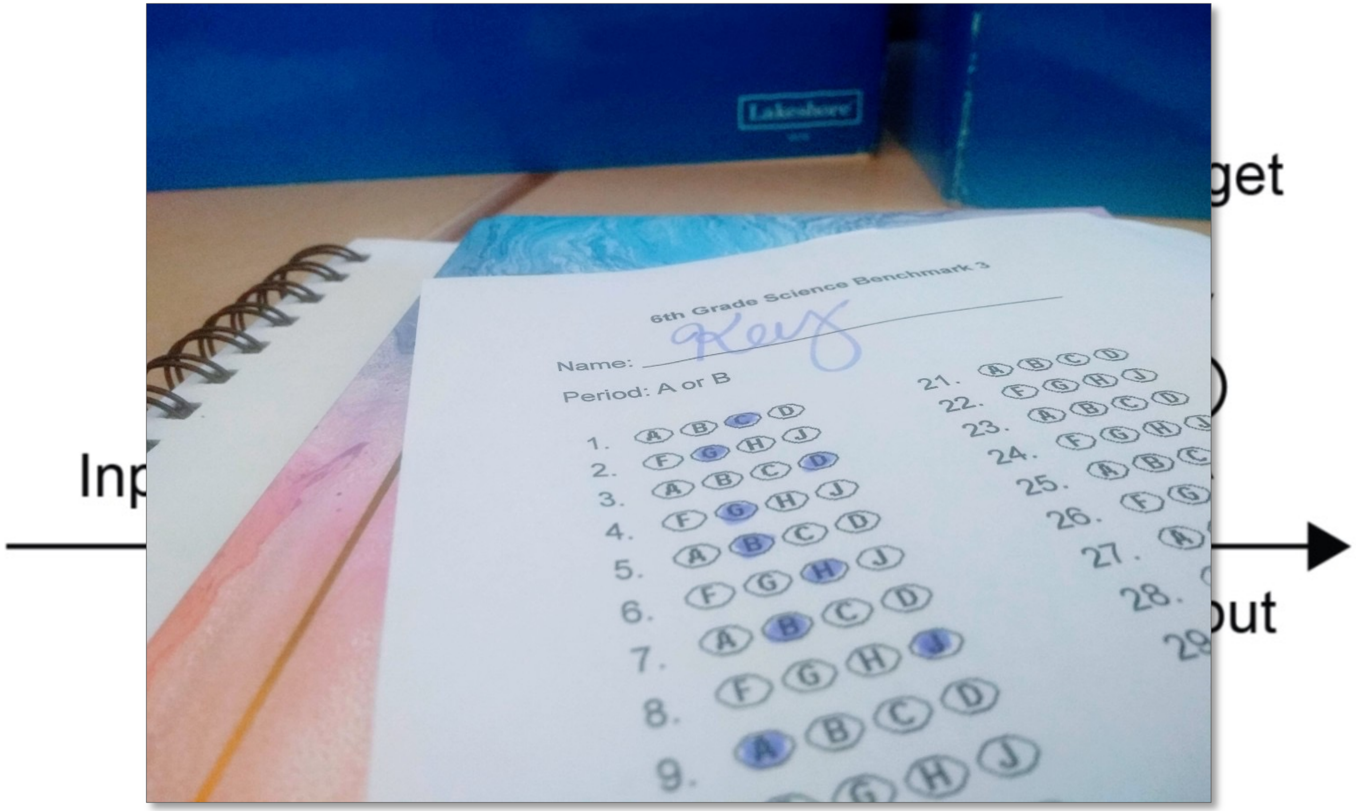


Image Credit: https://towardsdatascience.com/what-are-the-types-of-machine-learning-e2b9e5d1756f

Supervised Learning Logic



Supervised Learning Example

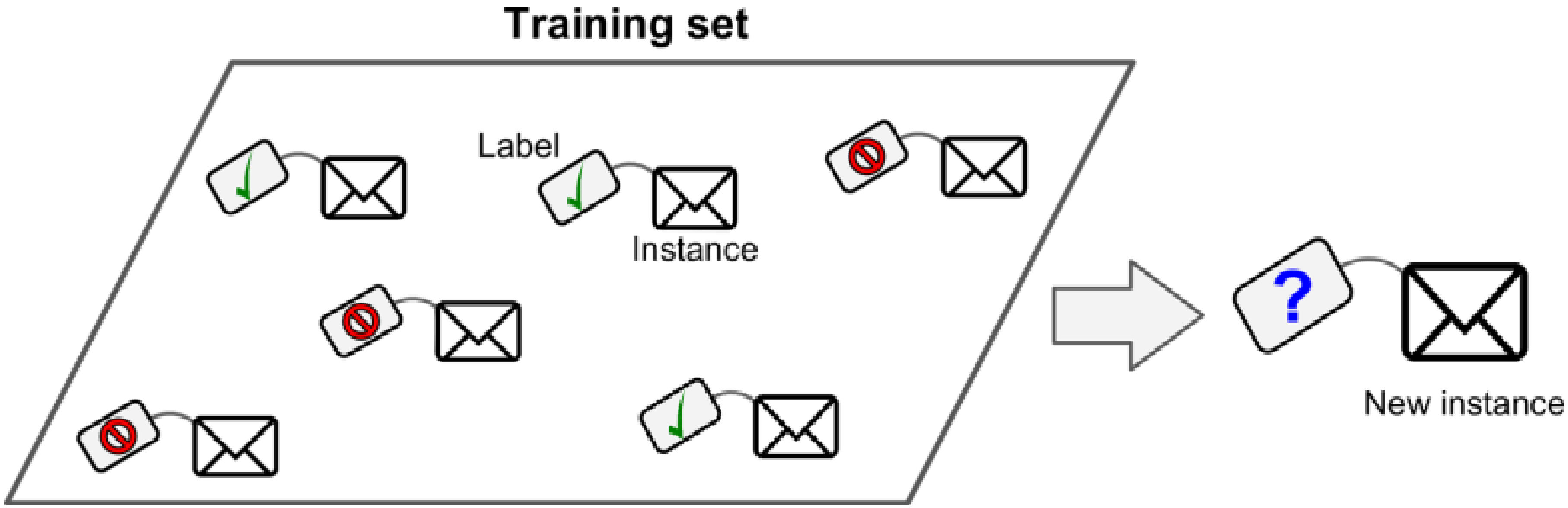
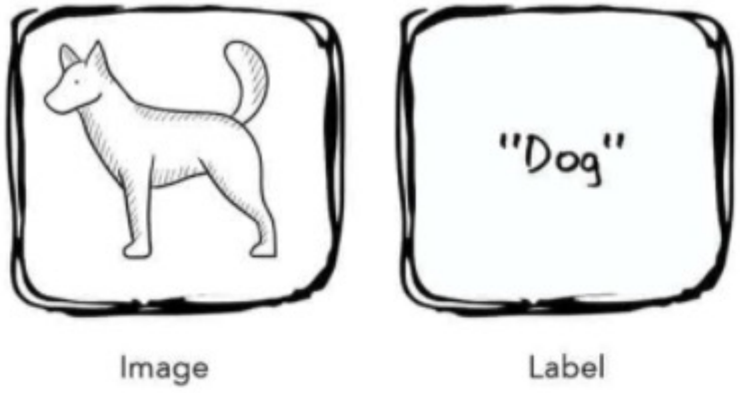


Figure 1-1. A labeled training set for spam classification.

Source: Geron, A. (2019). *Hands-On Machine Learning w/Scikit-Learn, Keras & Tensorflow* . Sebastopol, CA: O'Reilly Media, Inc. (Chapter 1)

Supervised Learning Example



Source: Taylor, M. (2017). *Neural Networks: A Visual Introduction for Beginners*. Sebastopol, CA: Blue Windmill Media.

Supervised Learning Example

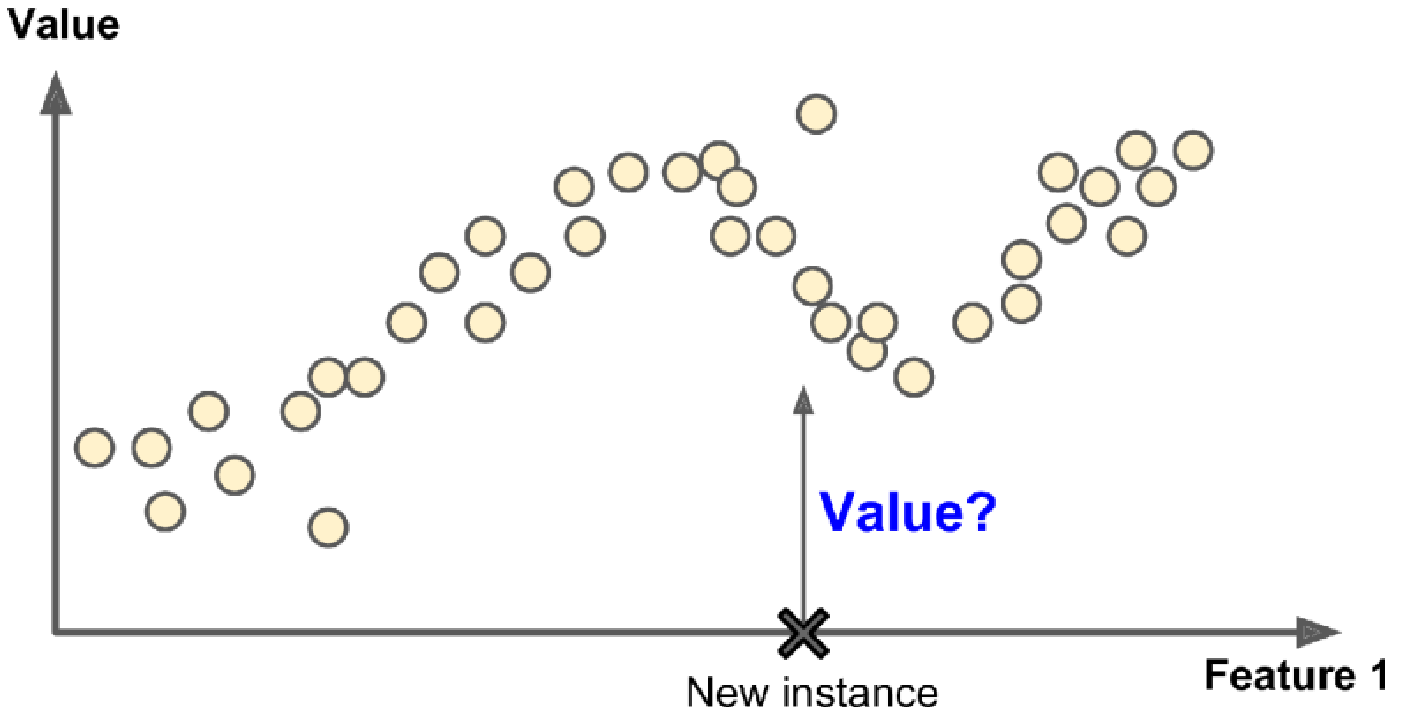
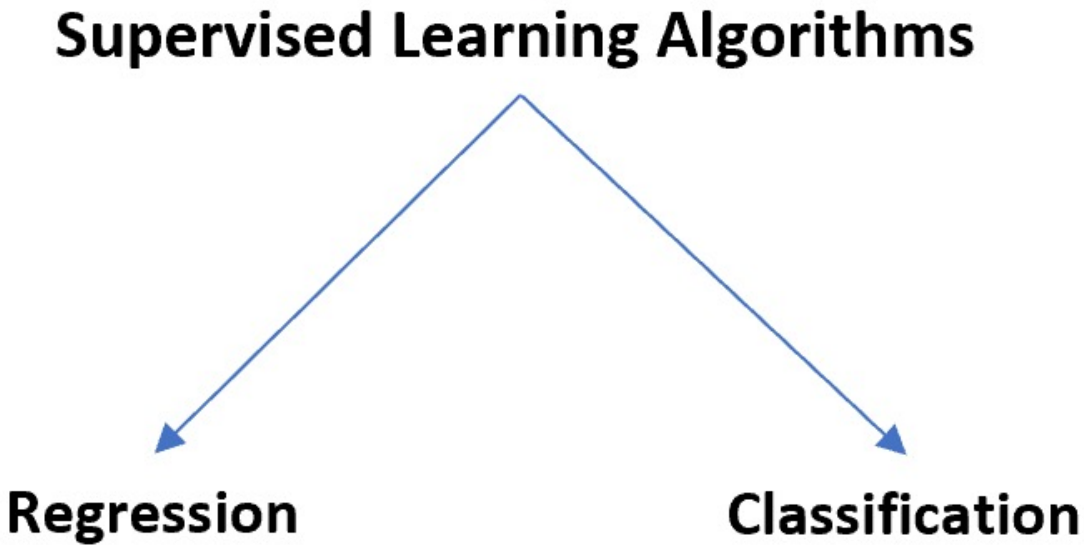
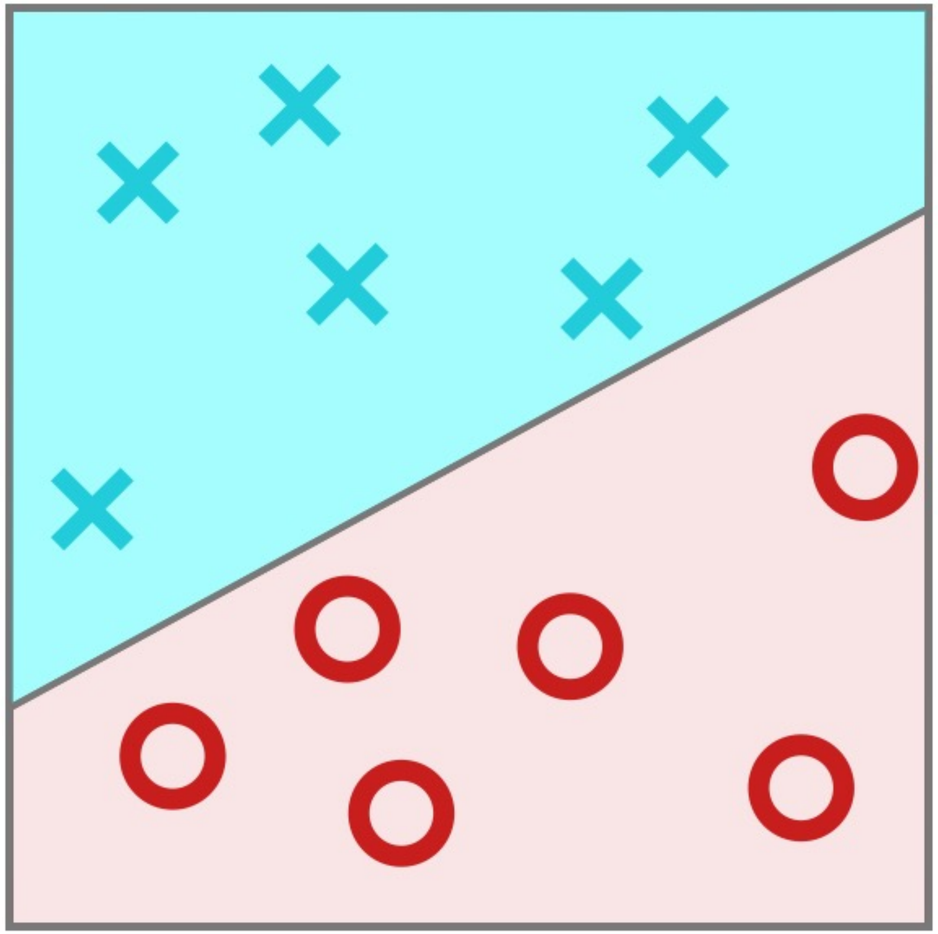


Figure 1-2. A regression problem: predict a value, given an input feature.

Source: Geron, A. (2019). *Hands-On Machine Learning w/Scikit-Learn, Keras & Tensorflow* . Sebastopol, CA: O'Reilly Media, Inc. (Chapter 1)

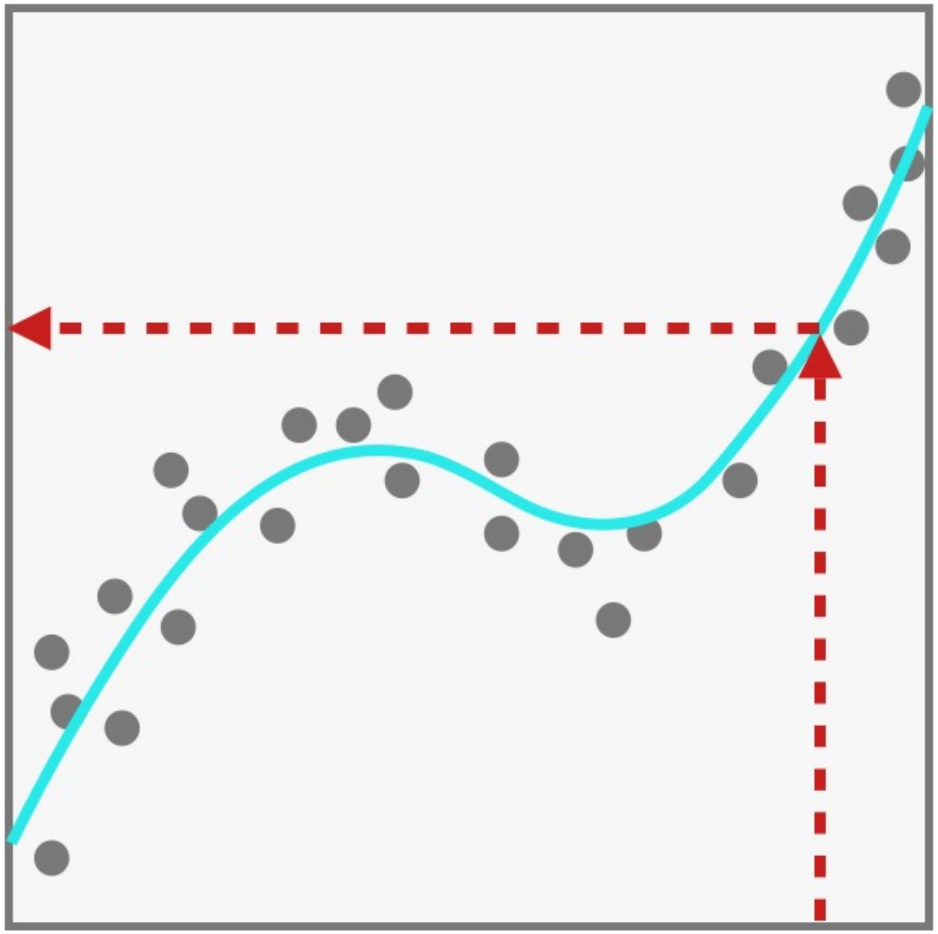

Supervised Learning Algorithms
Regression
Classification

Classification Groups observations into "classes"



Here, the line classifies the observations into X's and O's

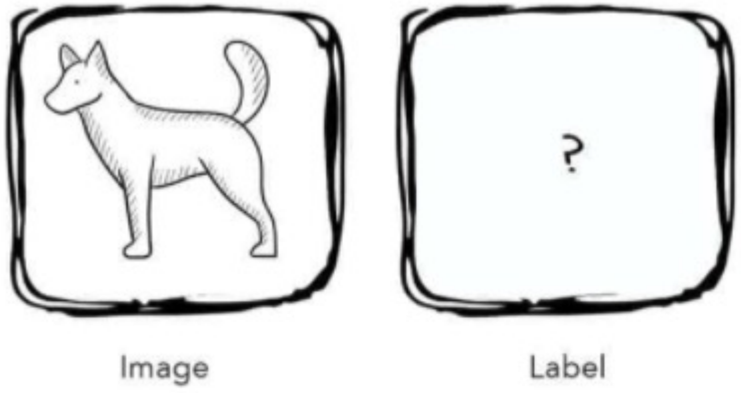
Regression predicts a numeric value



Here, the fitted line provides a predicted output, if we give it an input

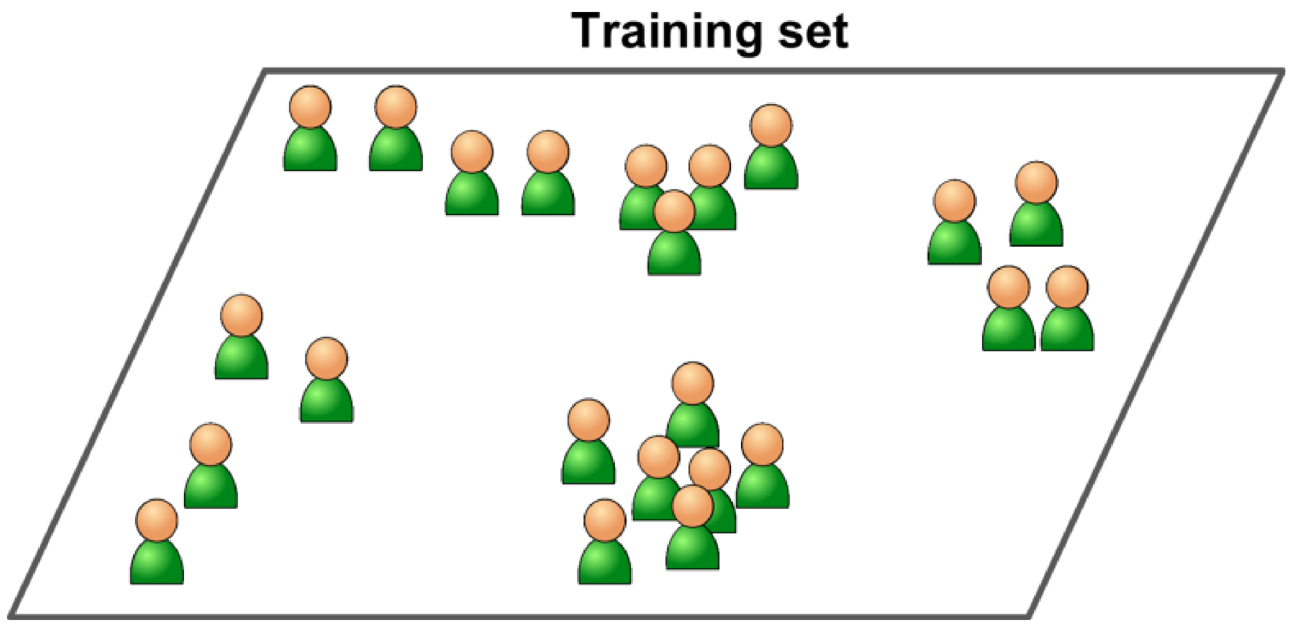
Unsupervised Learning Logic

Unsupervised Learning Example



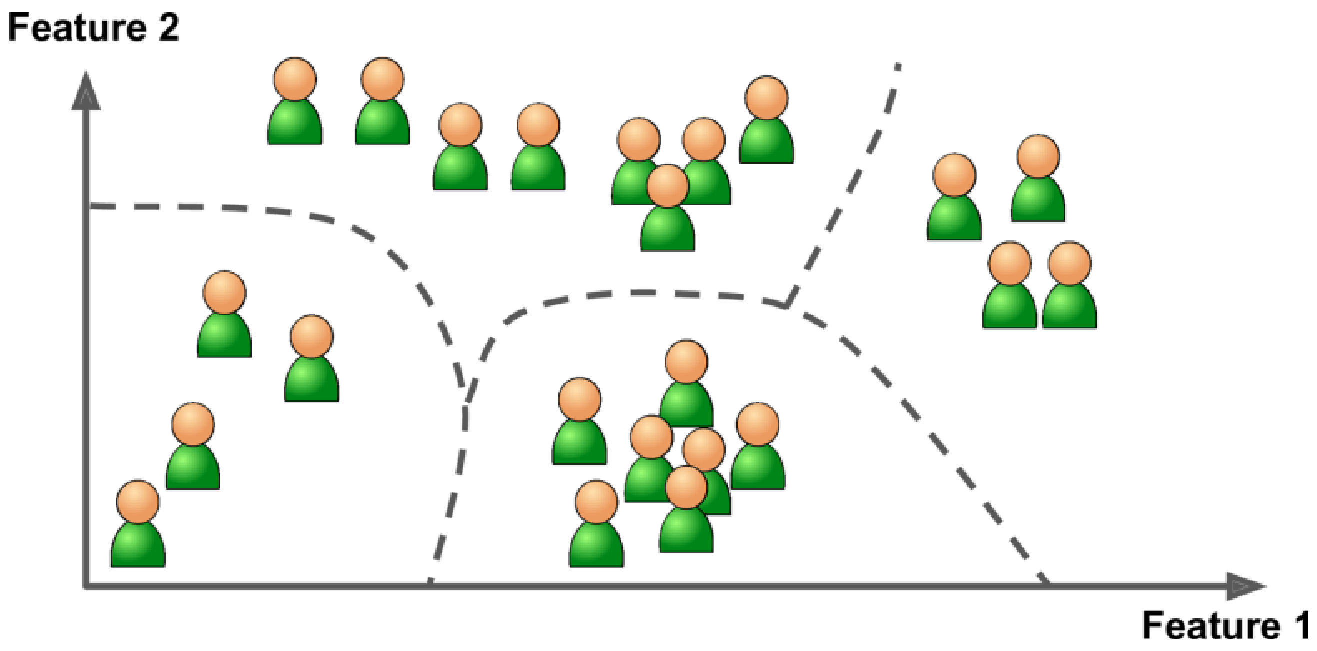
Source: Taylor, M. (2017). *Neural Networks: A Visual Introduction for Beginners*. Sebastopol, CA: Blue Windmill Media.

Unsupervised Learning Example



Source: Geron, A. (2019). *Hands-On Machine Learning w/Scikit-Learn, Keras & Tensorflow* . Sebastopol, CA: O'Reilly Media, Inc. (Chapter 1)

Unsupervised Learning Example



Source: Geron, A. (2019). *Hands-On Machine Learning w/Scikit-Learn, Keras & Tensorflow* . Sebastopol, CA: O'Reilly Media, Inc. (Chapter 1)

Unsupervised Learning Example

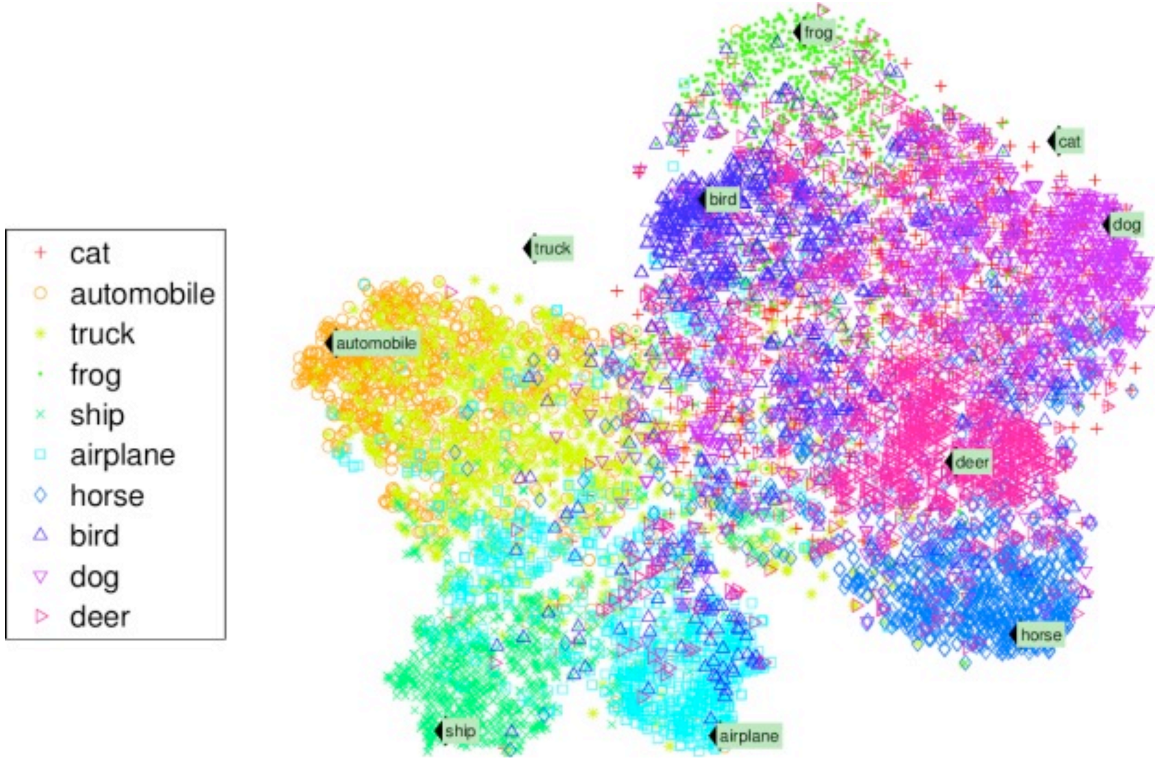
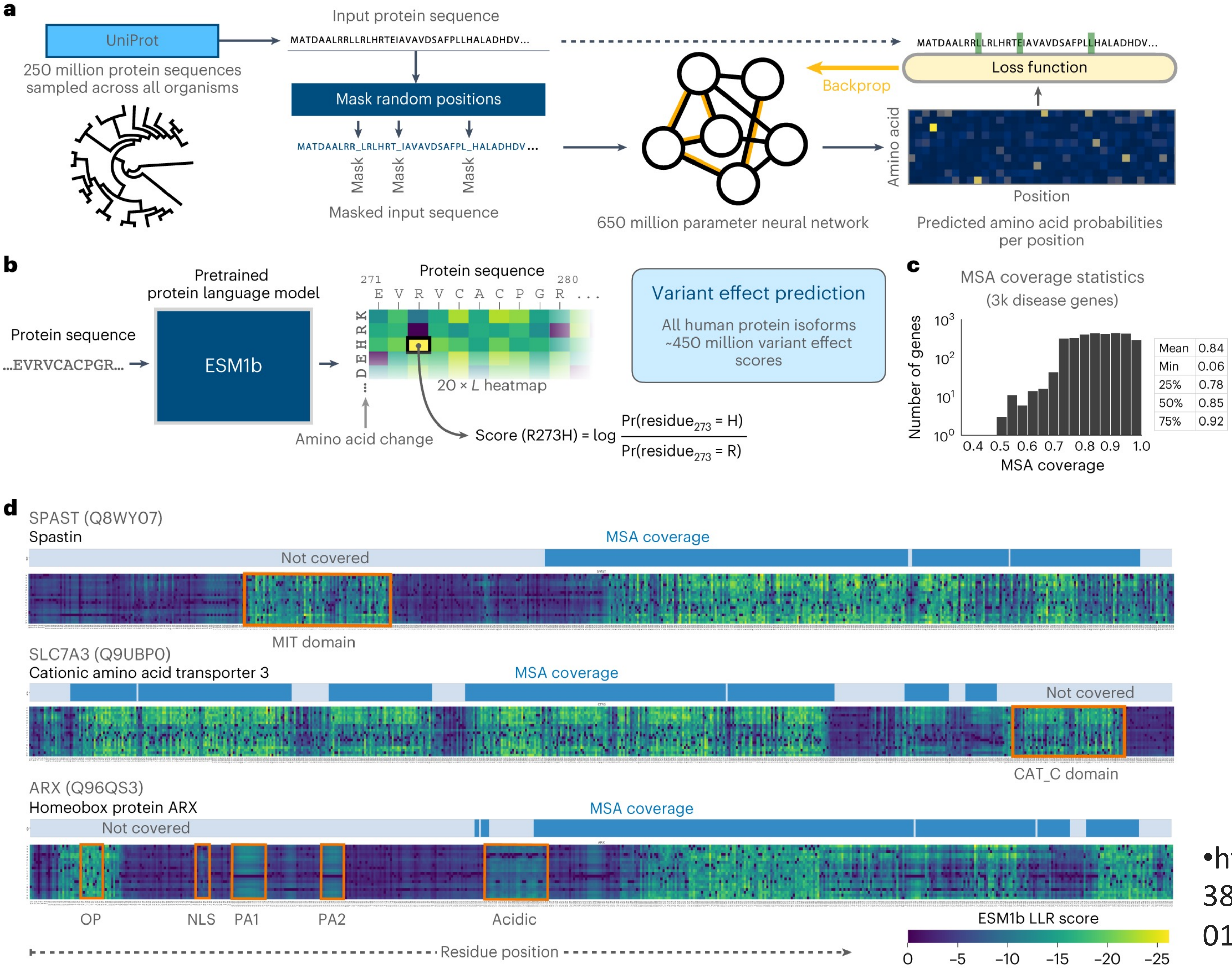


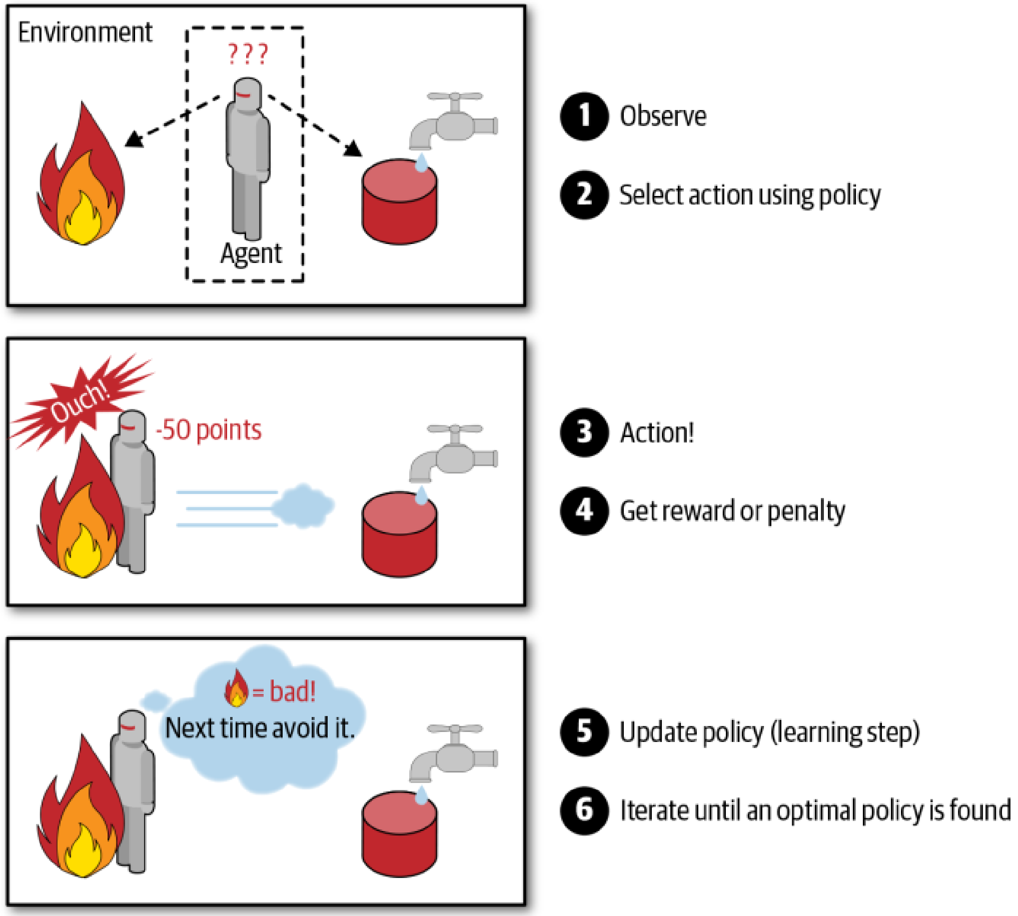
Figure 1-6. Data visualization.



•https://doi.org/10.1038/s41588-023-01465-0

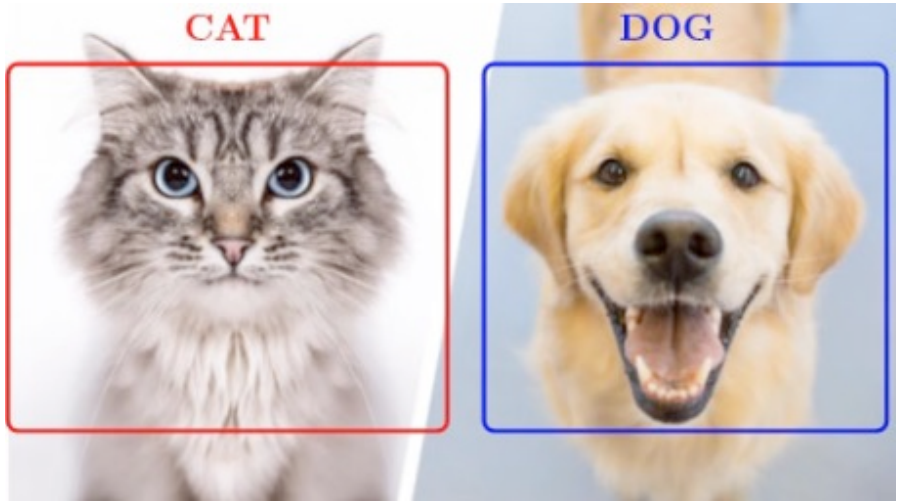
Reinforcement Learning Logic

Reinforcement Learning Example



Source: Geron, A. (2019). *Hands-On Machine Learning w/Scikit-Learn, Keras & Tensorflow* . Sebastopol, CA: O'Reilly Media, Inc. (Chapter 1)

Detecting and Classifying Dogs and Cats in an Image

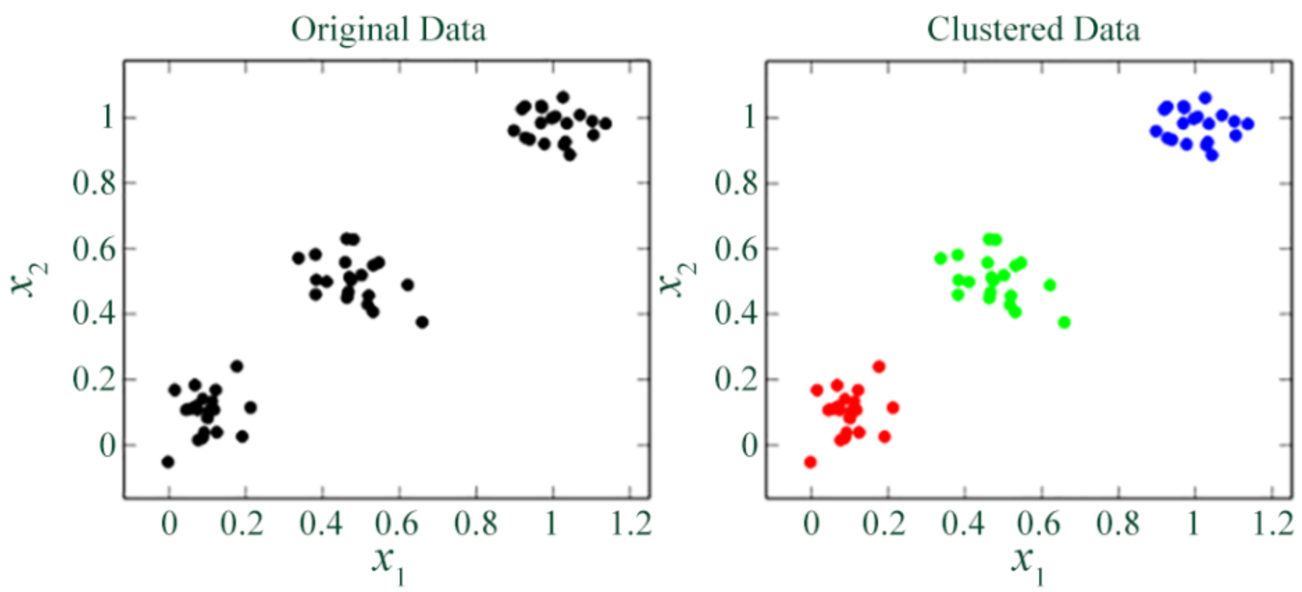


What kind of learning is it?

Supervised | Unsupervised | Reinforcement

Source: Palmas, A. (2019). *The Reinforcement Learning Workshop*. Birmingham, UK: Packt Publishing. (Chapter 1)

Identifying Clusters in a Dataset



What kind of learning is it?

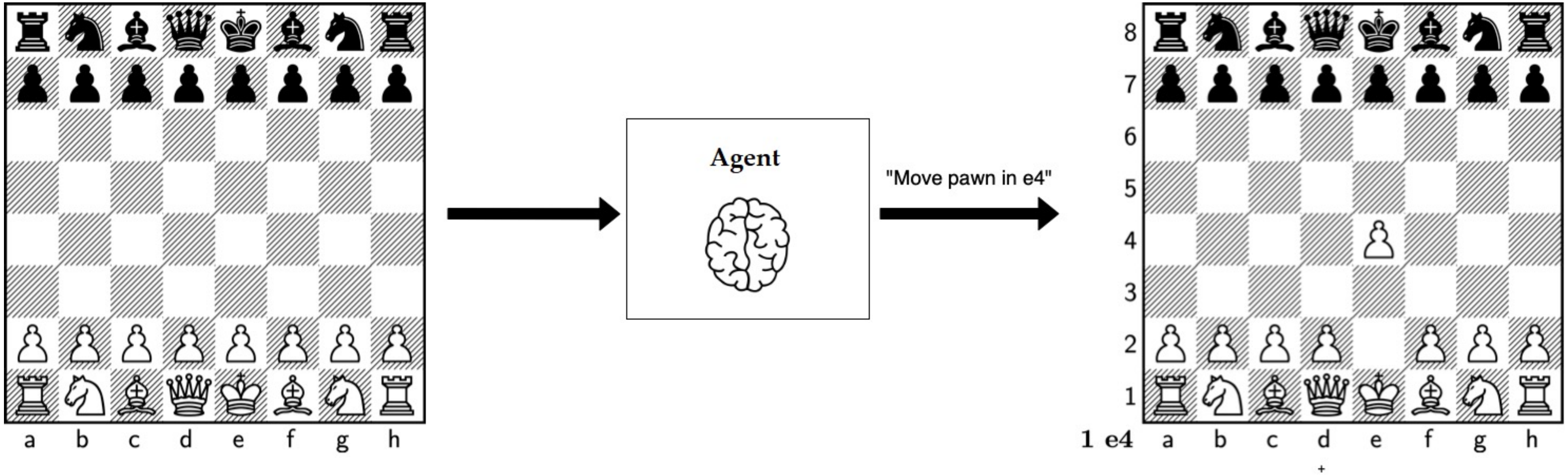
Supervised

Unsupervised

Reinforcement

Source: Palmas, A. (2019). *The Reinforcement Learning Workshop*. Birmingham, UK: Packt Publishing. (Chapter 1)

Playing Chess



What kind of learning is it?

Supervised

Unsupervised

Reinforcement

Source: Palmas, A. (2019). *The Reinforcement Learning Workshop*. Birmingham, UK: Packt Publishing. (Chapter 1)

Predicting cancer risk based genomic sequences

What kind of learning is it?

Supervised | Unsupervised | Reinforcement

Source: Palmas, A. (2019). *The Reinforcement Learning Workshop*. Birmingham, UK: Packt Publishing. (Chapter 1)

Applications

Supervised Learning

A. Classification
- Fraud Detection
- Image Classification
- Customer Retention
- Diagnostics

B. Regression
- Forecasting
- Predictions
- Process Optimization
- New insights

Reinforcement Learning

A. Robot Navigation
- Skill Acquisition
- Learning Tasks
- Game AI
- Real-Time Decisions

Unsupervised Learning

A. Dimension Reduction
- Feature Elicitation
- Structure Discovery
- Data Compression
- Big Data Visualization

B. Clustering
- Recommender Systems
- Targeted Marketing
- Customer Segmentation

The supervised learning paradigm

Cat or Dog?

Mathematical function mapping image to label ("cat" or "dog")

cat

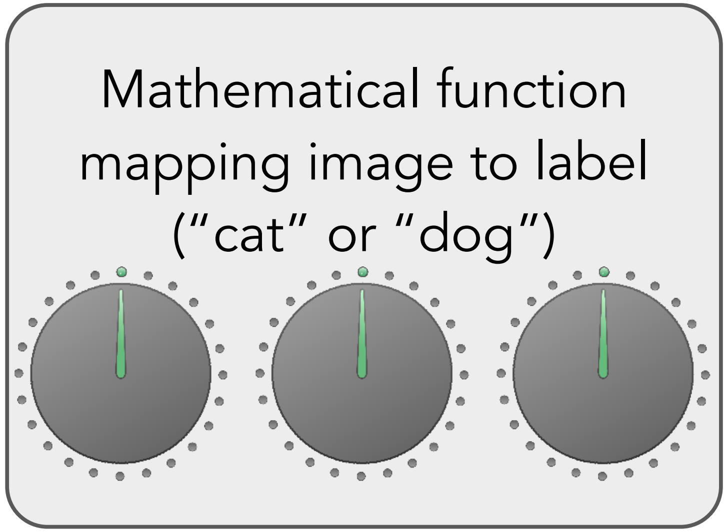
Cat or Dog?

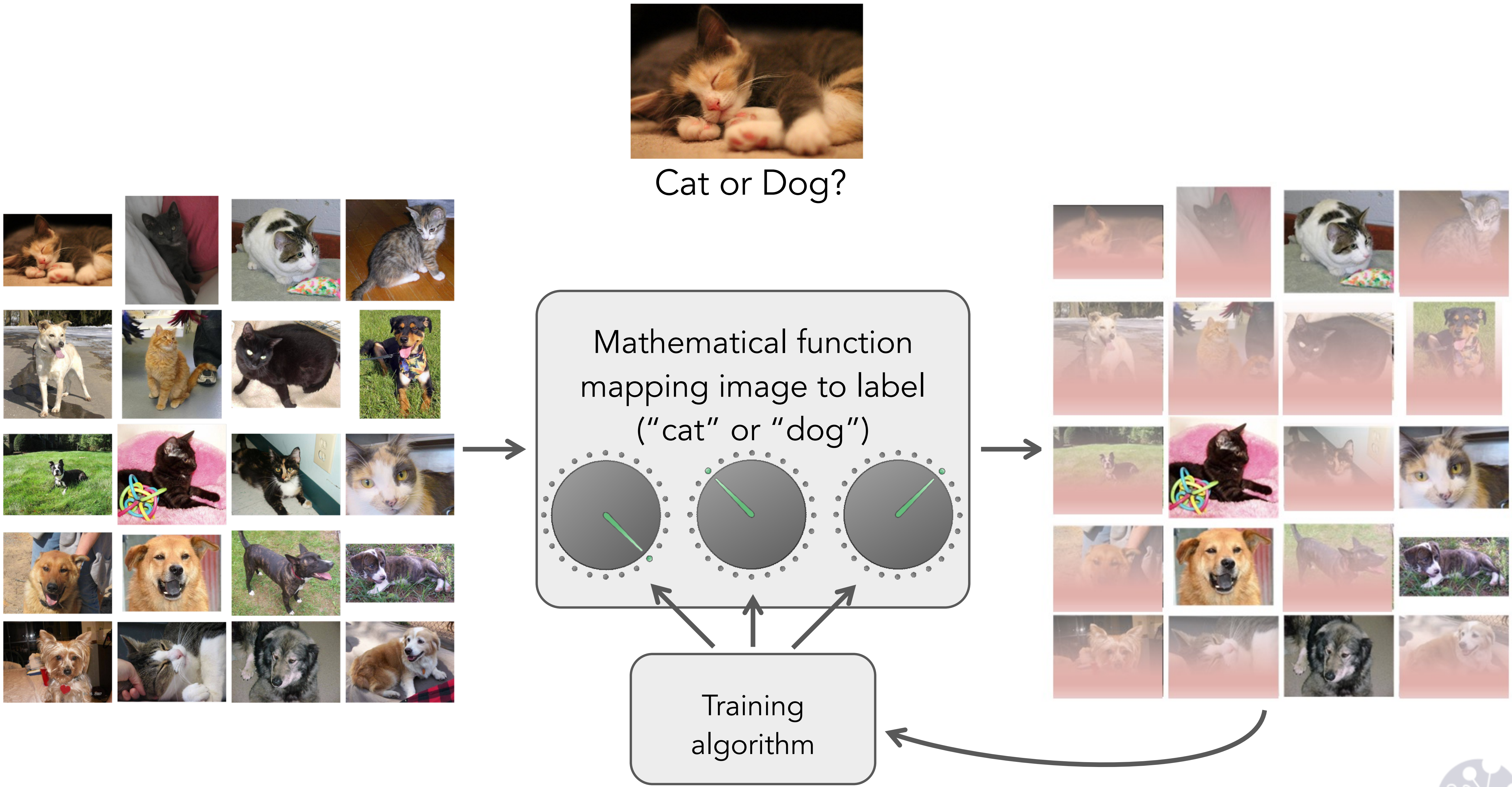
Mathematical function
mapping image to label
(“cat” or “dog”)

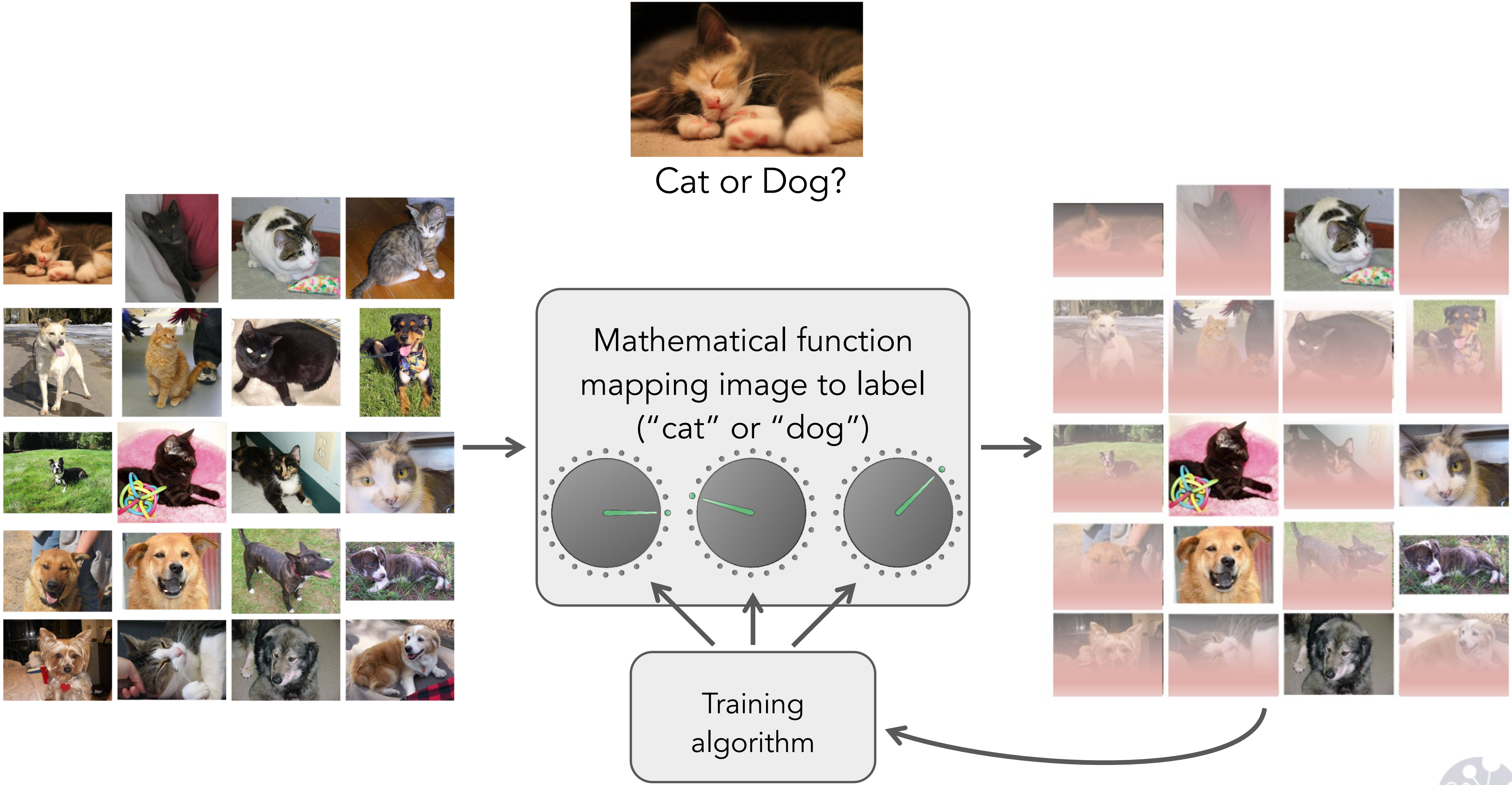
E.g.: $y = mx + b$

Special numerical values in the
function are “knobs” we can turn
to adjust the outputs.

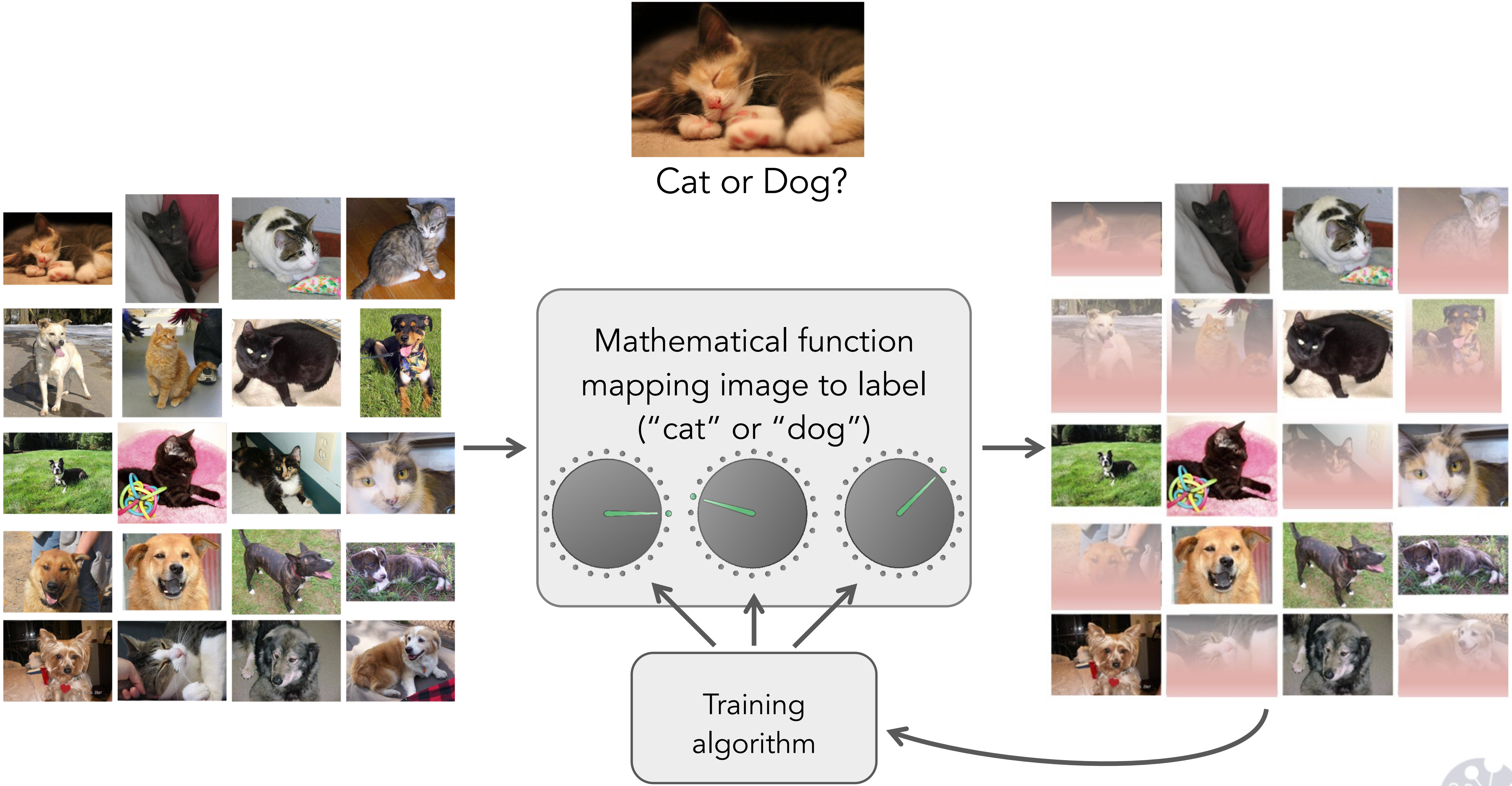
Cat or Dog?

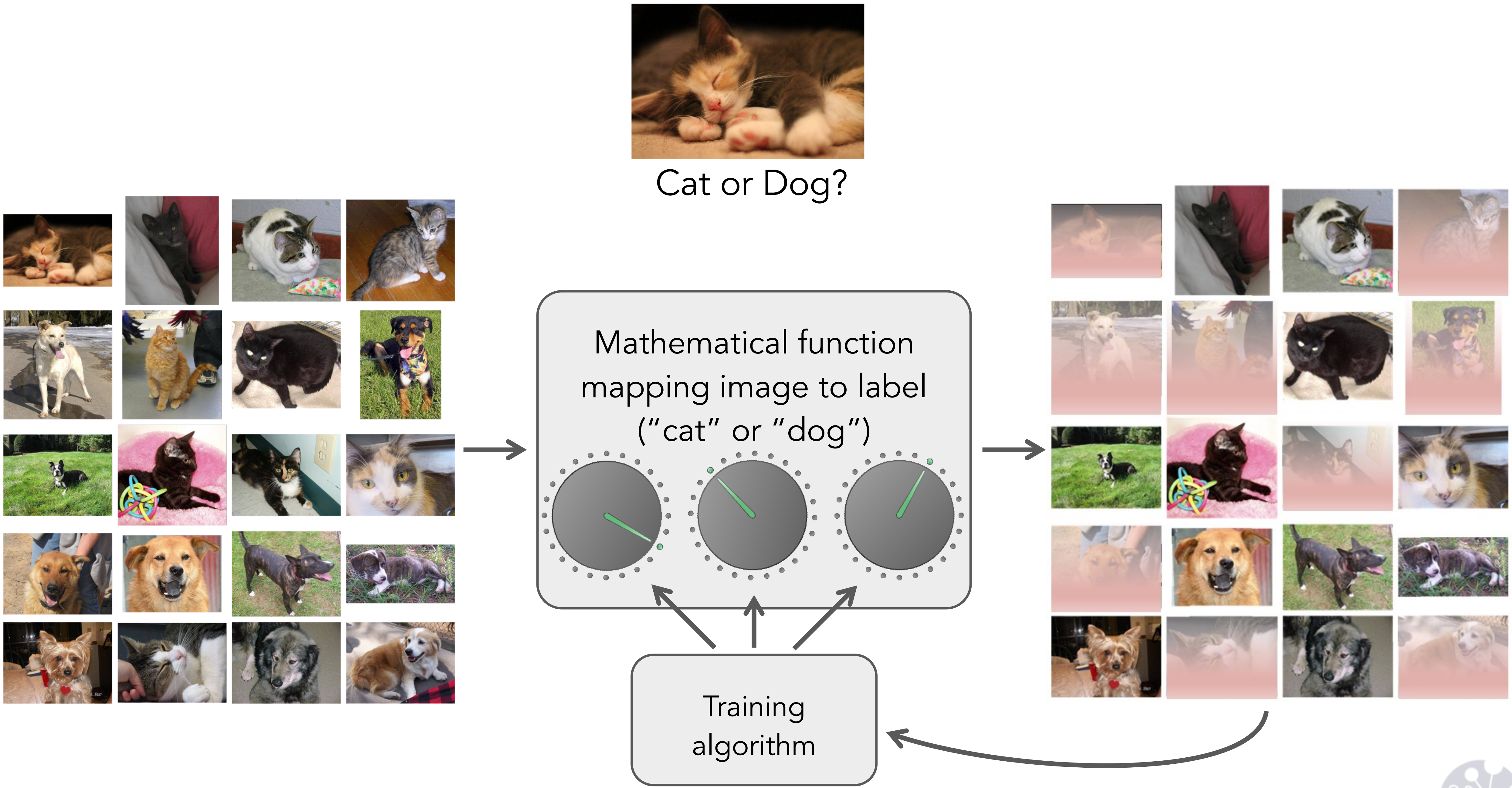


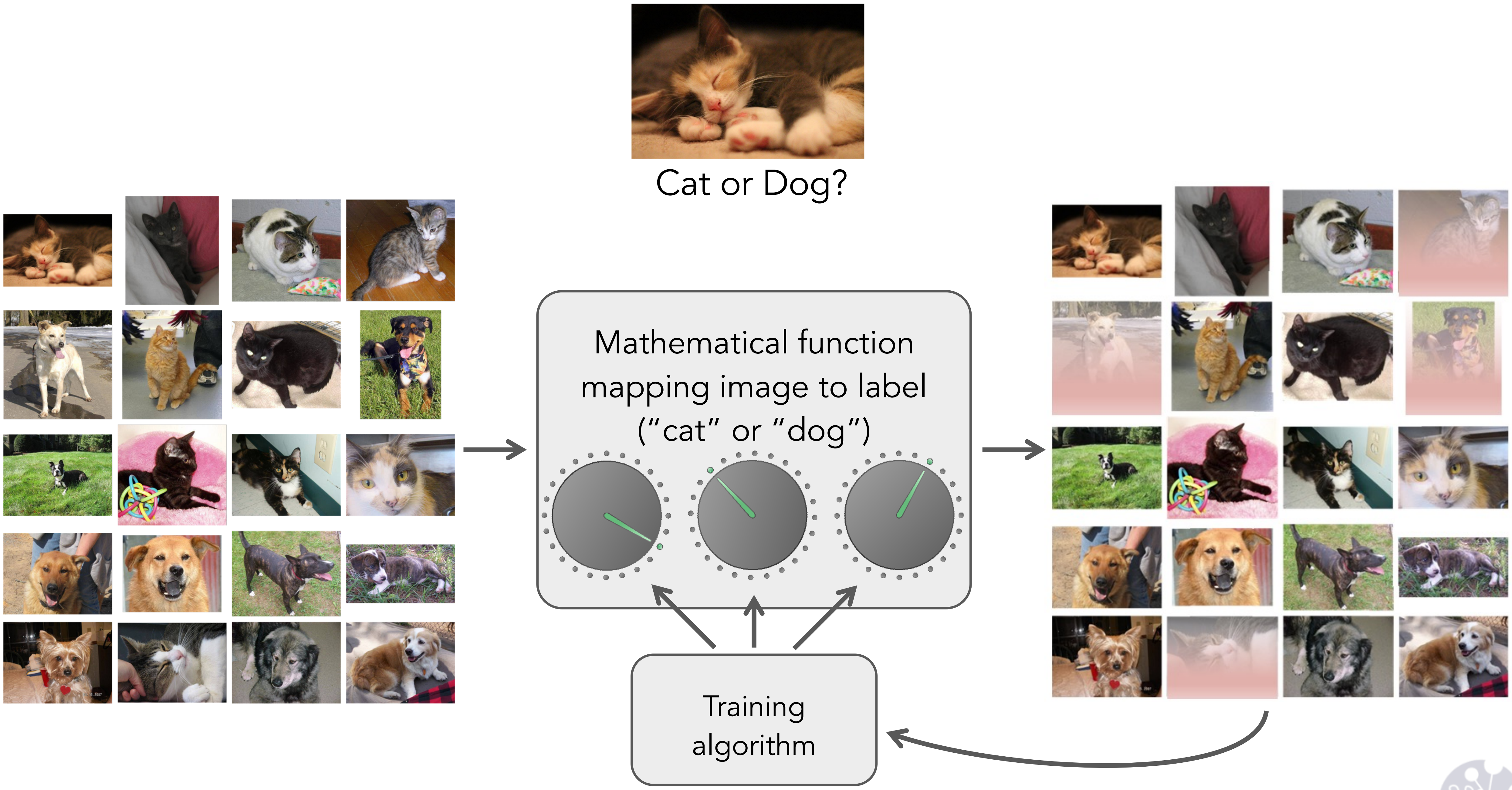

Cat or Dog?
Mathematical function mapping image to label ("cat" or "dog")
Training algorithm



Cat or Dog?
Mathematical function
mapping image to label
("cat" or "dog")
Training
algorithm


Cat or Dog?
Mathematical function
mapping image to label
("cat" or "dog")
Training
algorithm


Cat or Dog?
Mathematical function mapping image to label ("cat" or "dog")
Training algorithm

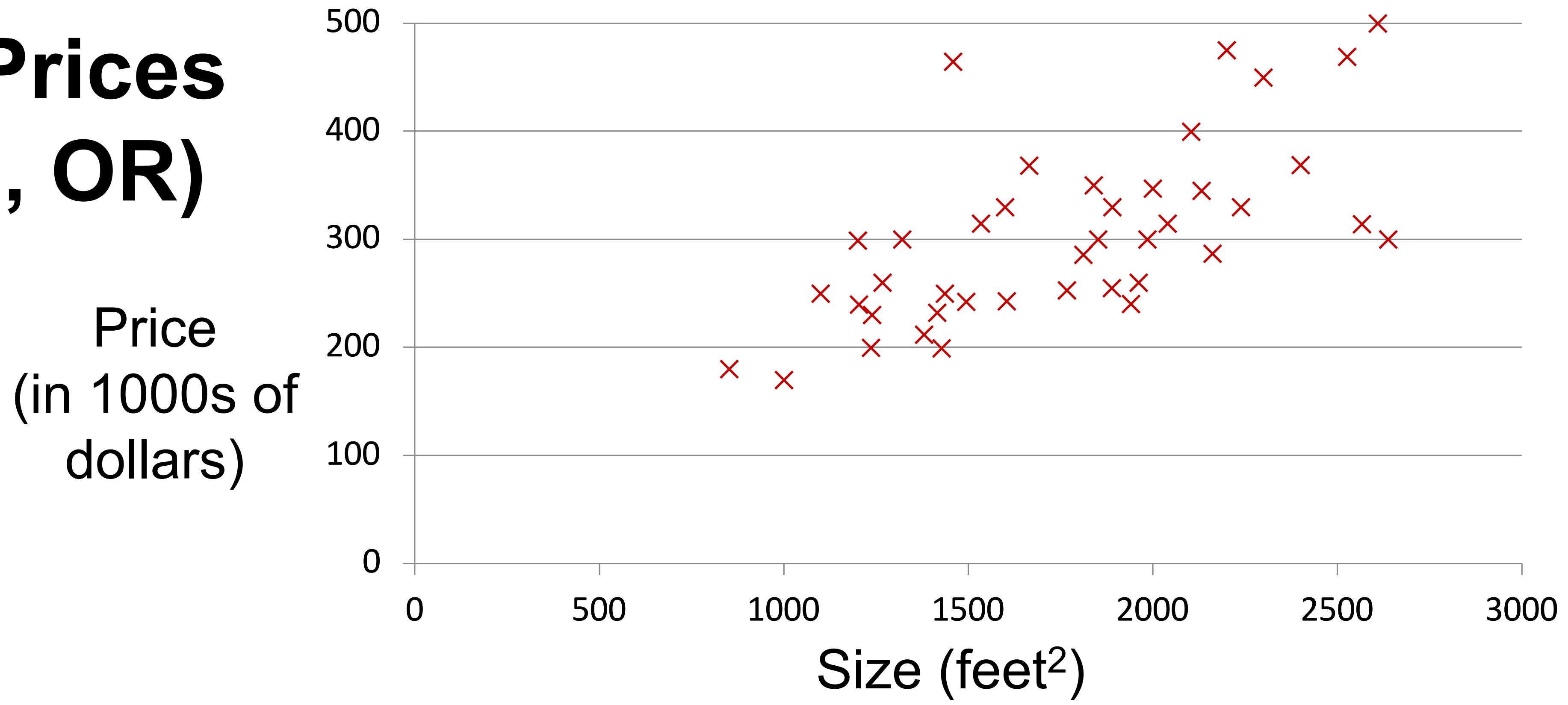


Cat or Dog?
Mathematical function mapping image to label ("cat" or "dog")
Training algorithm

Simplest supervised learning
- linear regression with one variable

Task: Fit a model to predict house prices

Housing Prices (Portland, OR)



<u>Supervised Learning</u>

Given the “right answer” for each example in the data.

<u>Regression Problem</u>

Predict real-valued output

Training set of housing prices (Portland, OR)

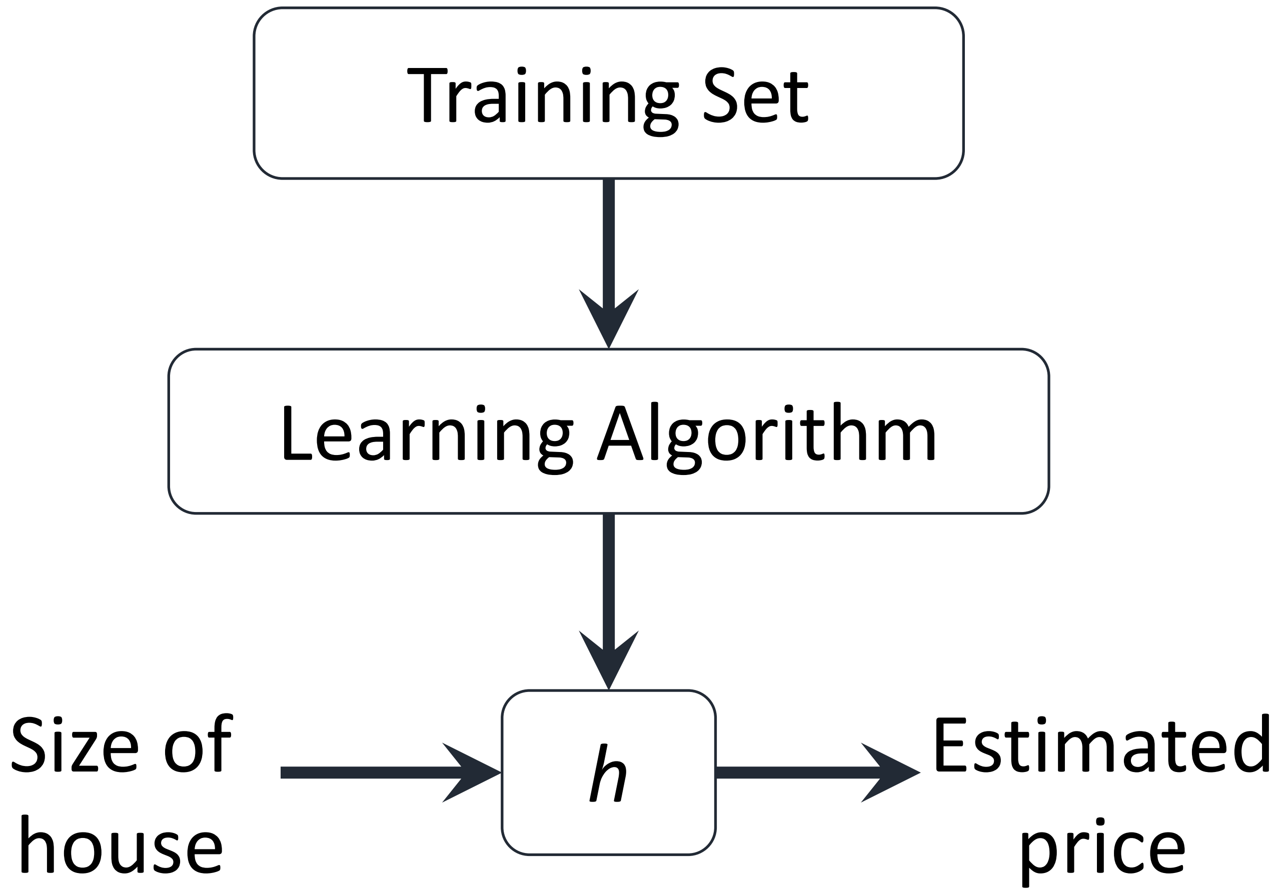
Size in feet^2 (x)	Price ($) in 1000's (y)
2104	460
1416	232
1534	315
852	178
…	…

Notation:

m = Number of training examples

x’s = “input” variable / features

y’s = “output” variable / “target” variable



How do we represent *h* ?

$$h_\theta(x) = \theta_0 + \theta_1 x$$

Linear regression with one variable.
Univariate linear regression.

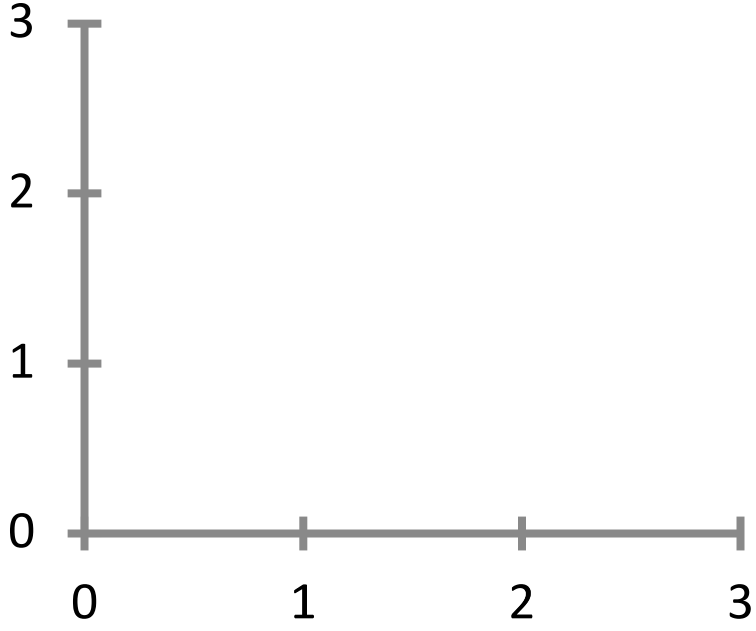
Training Set

Size in feet^2 (x)	**Price ($) in 1000's (y)**
2104	460
1416	232
1534	315
852	178
…	…

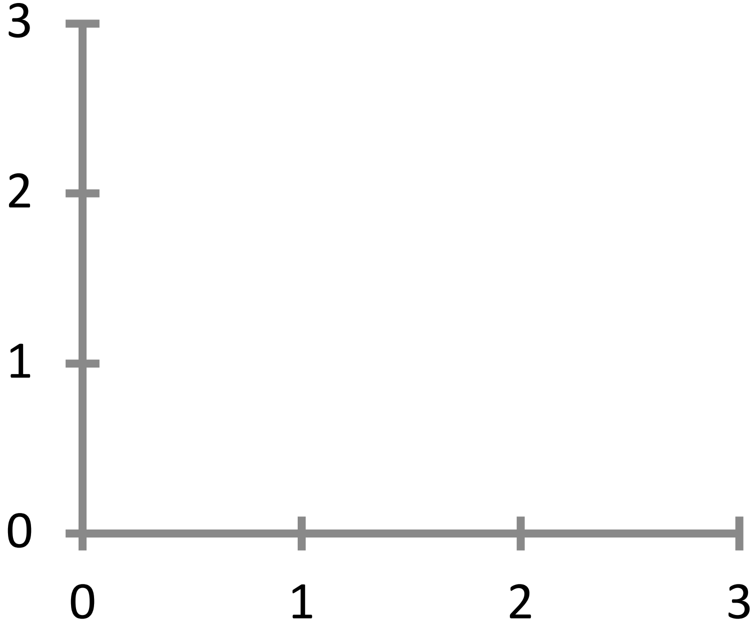
Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$

θ_i's: Parameters

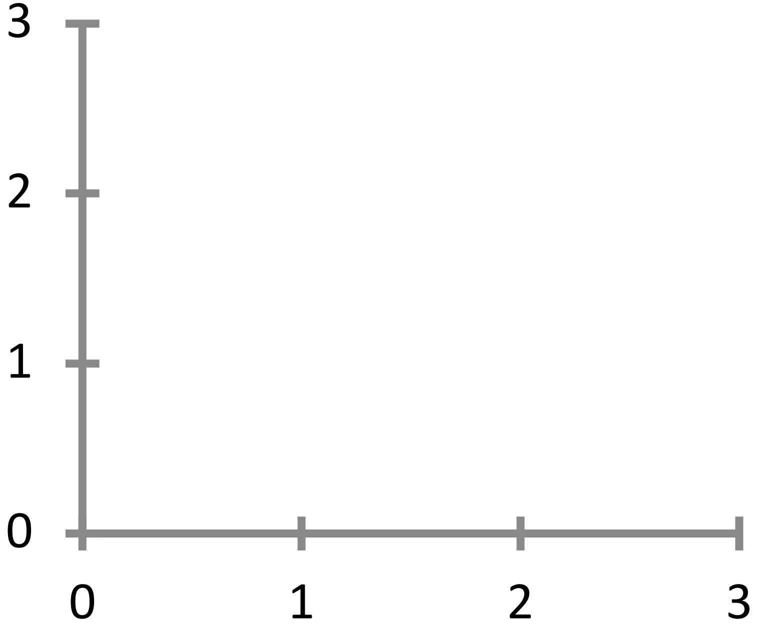
$$h_\theta(x) = \theta_0 + \theta_1 x$$



$$\theta_0 = 1.5$$
$$\theta_1 = 0$$

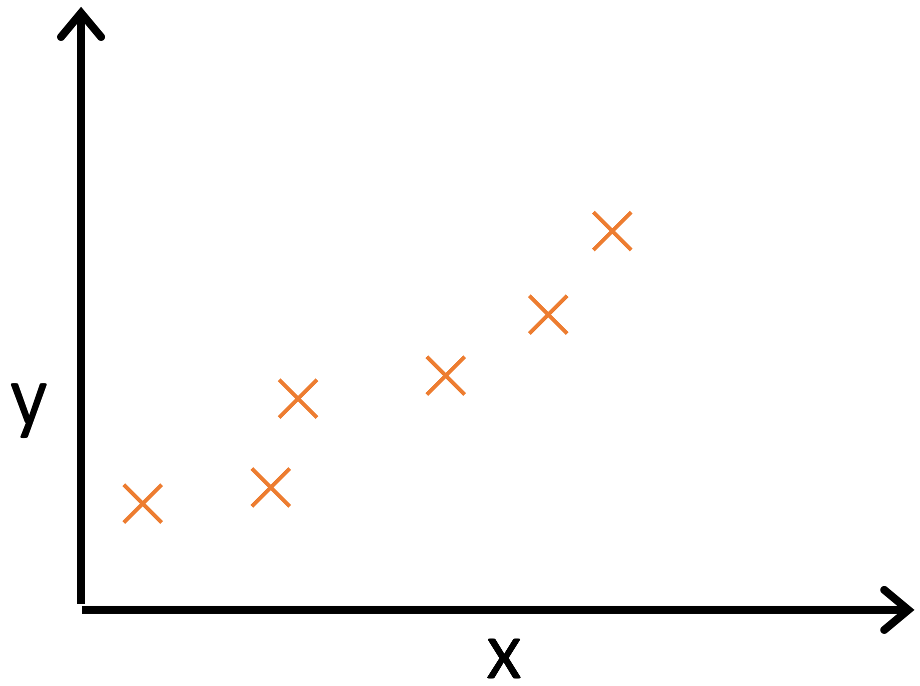


$$\theta_0 = 0$$
$$\theta_1 = 0.5$$



$$\theta_0 = 1$$
$$\theta_1 = 0.5$$

How to choose the model parameters?

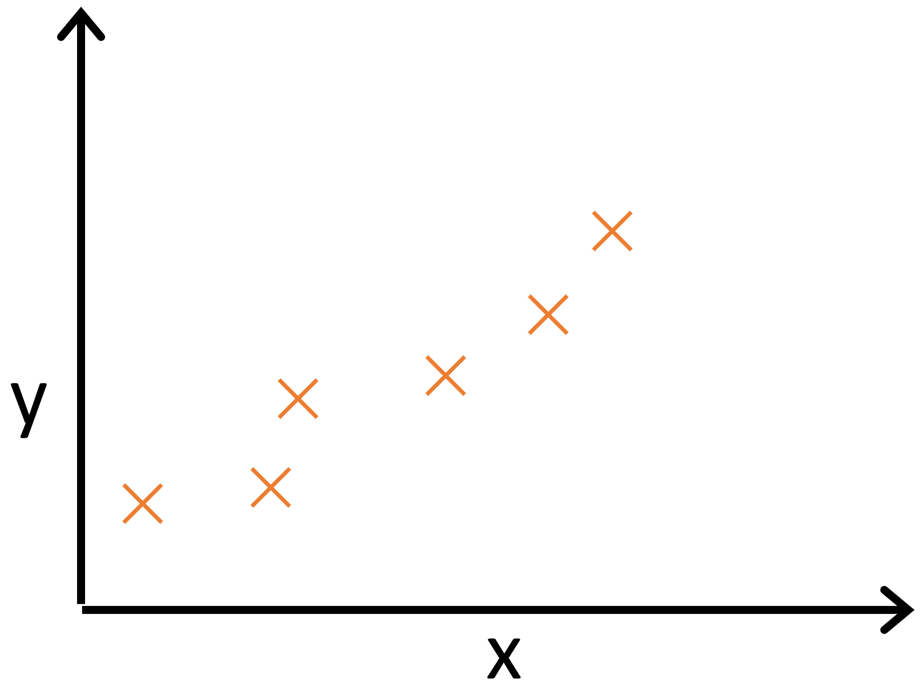


Idea: Choose θ_0, θ_1 so that $h_\theta(x)$ is close to for our training examples

Cost function:

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

Introducing the cost function



Cost function measures how well the model fits the training data given the model parameters

Mean squared error **(MSE)**

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

Measures the mean squared distance between the true labels and the predictions

Simplified

Hypothesis:

$$h_\theta(x) = \theta_0 + \theta_1 x$$

$$h_\theta(x) = \theta_1 x$$

Parameters:

$$\theta_0, \theta_1$$

$$\theta_1$$

Cost Function:

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

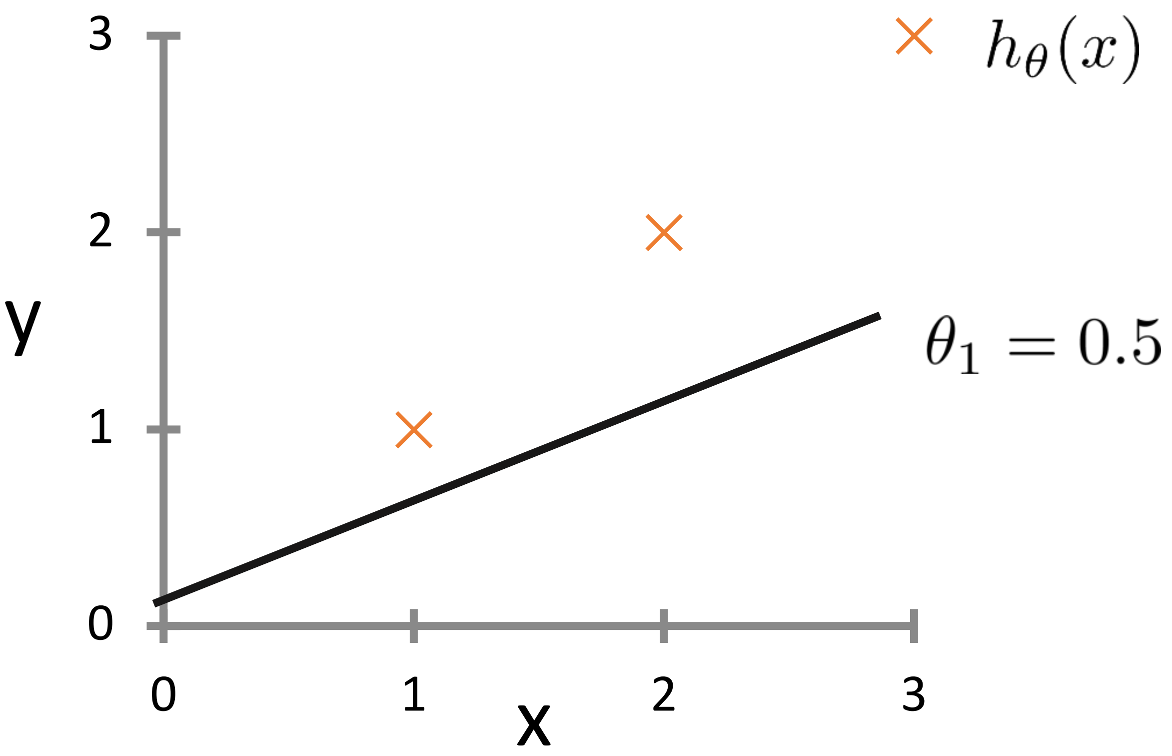
$$J(\theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

Goal: $\underset{\theta_0, \theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$

$$\underset{\theta_1}{\text{minimize}}\ J(\theta_1)$$

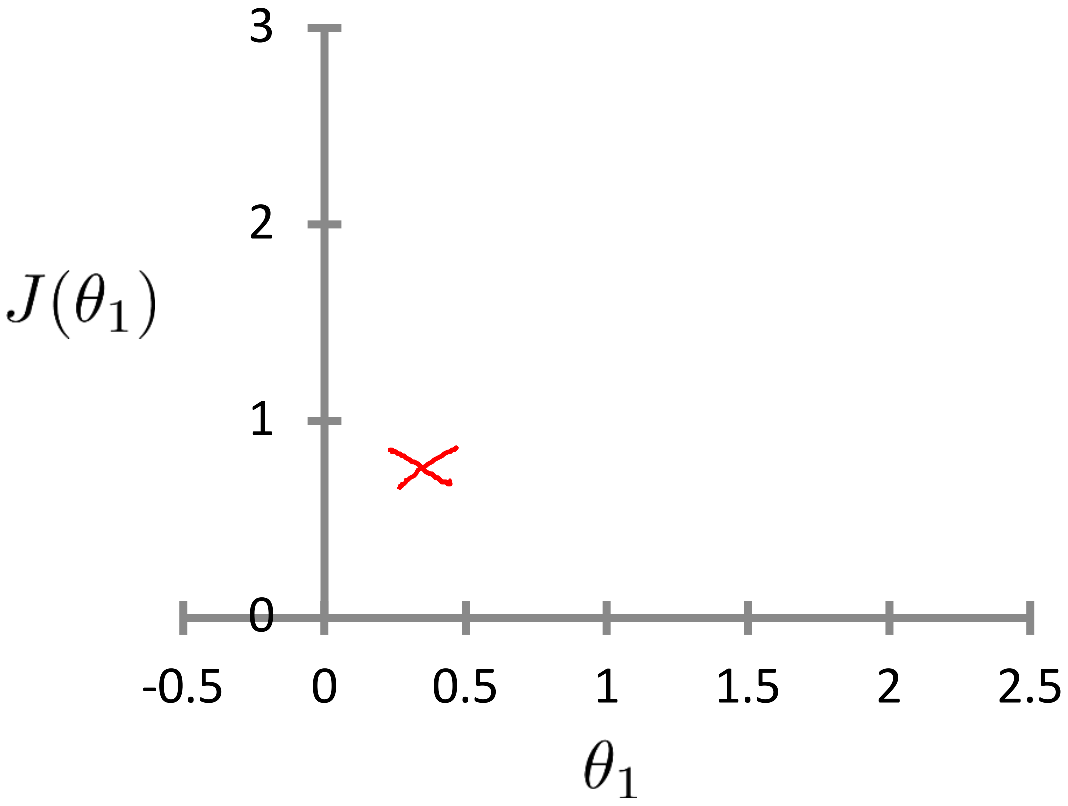
$$h_\theta(x)$$

(for fixed θ_1, this is a function of x)



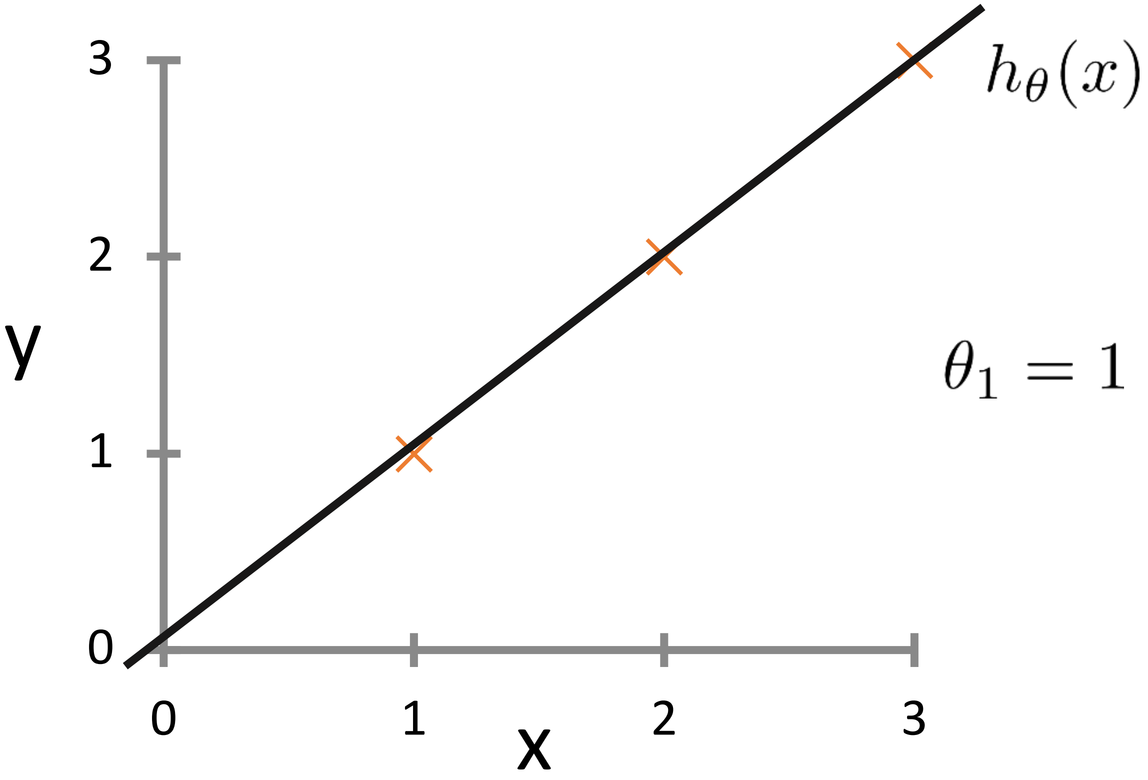
$$J(\theta_1)$$

(function of the parameter θ_1)



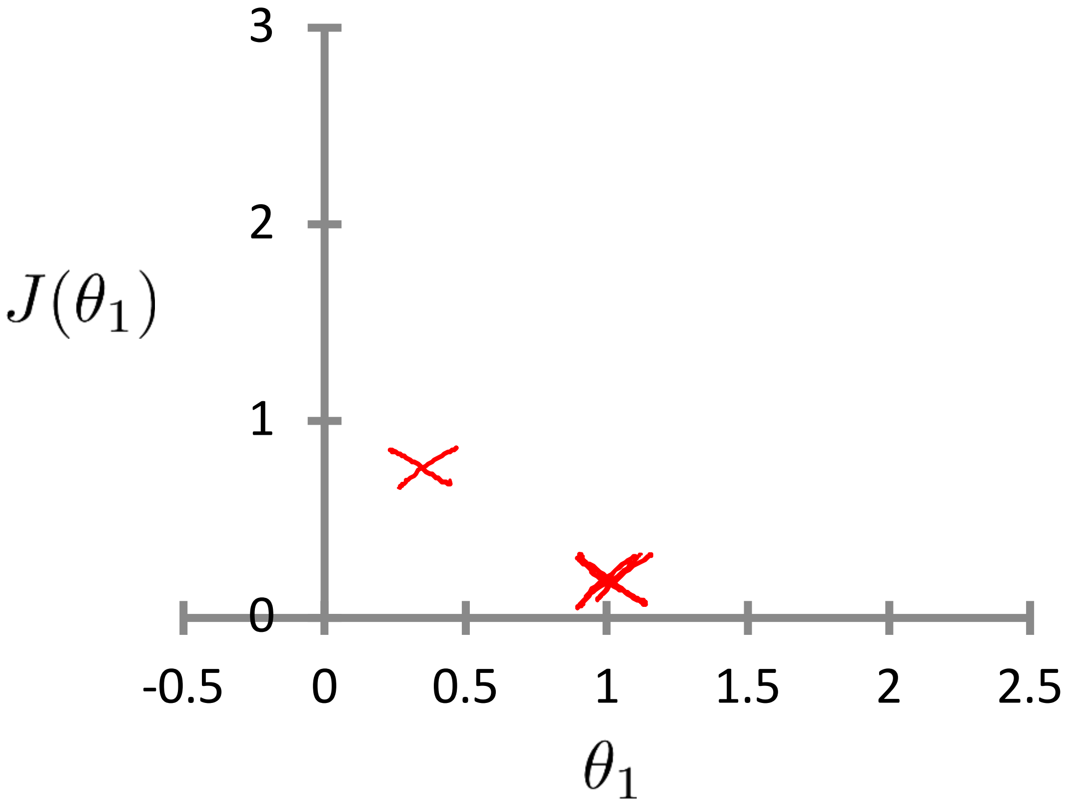
$h_\theta(x)$

(for fixed θ_1, this is a function of x)



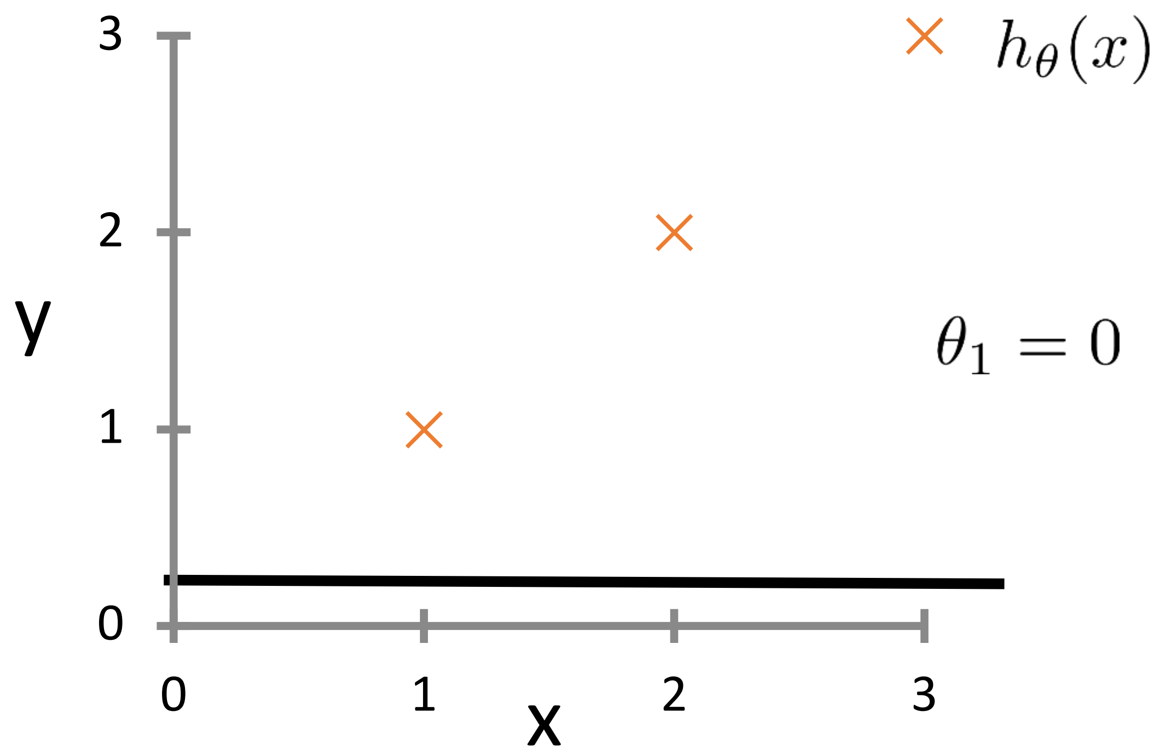
$J(\theta_1)$

(function of the parameter θ_1)



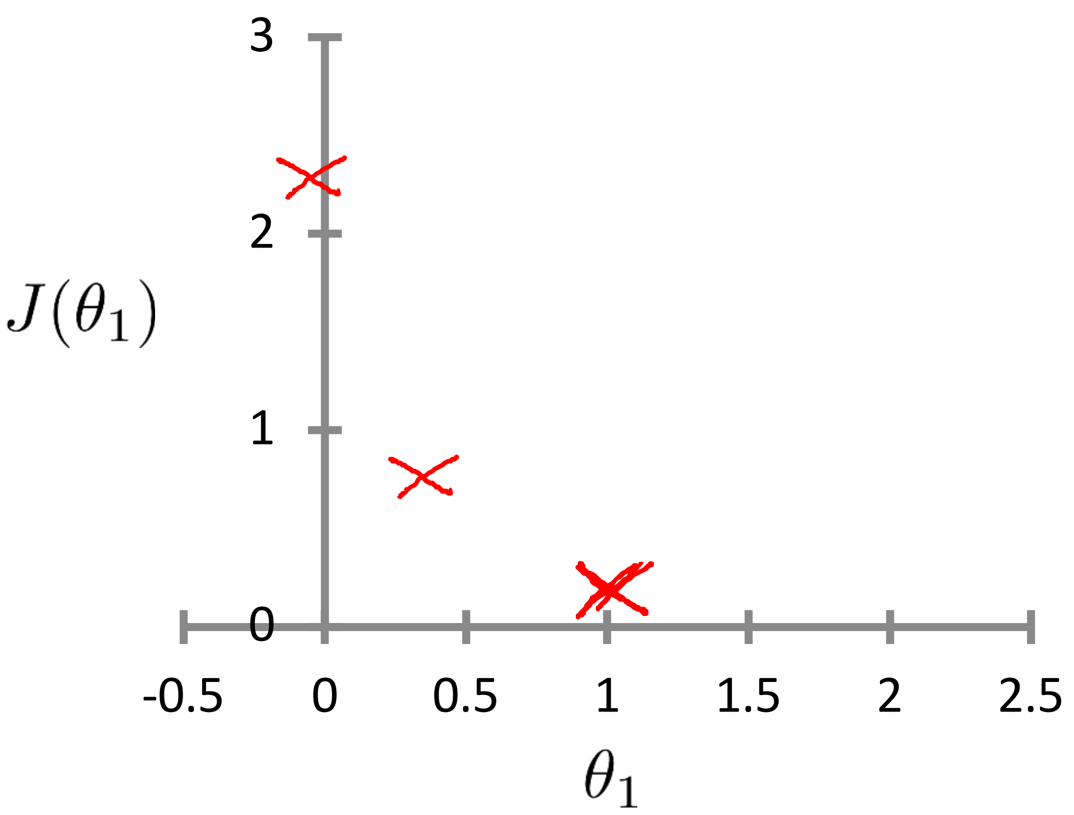
$$h_\theta(x)$$

(for fixed θ_1, this is a function of x)



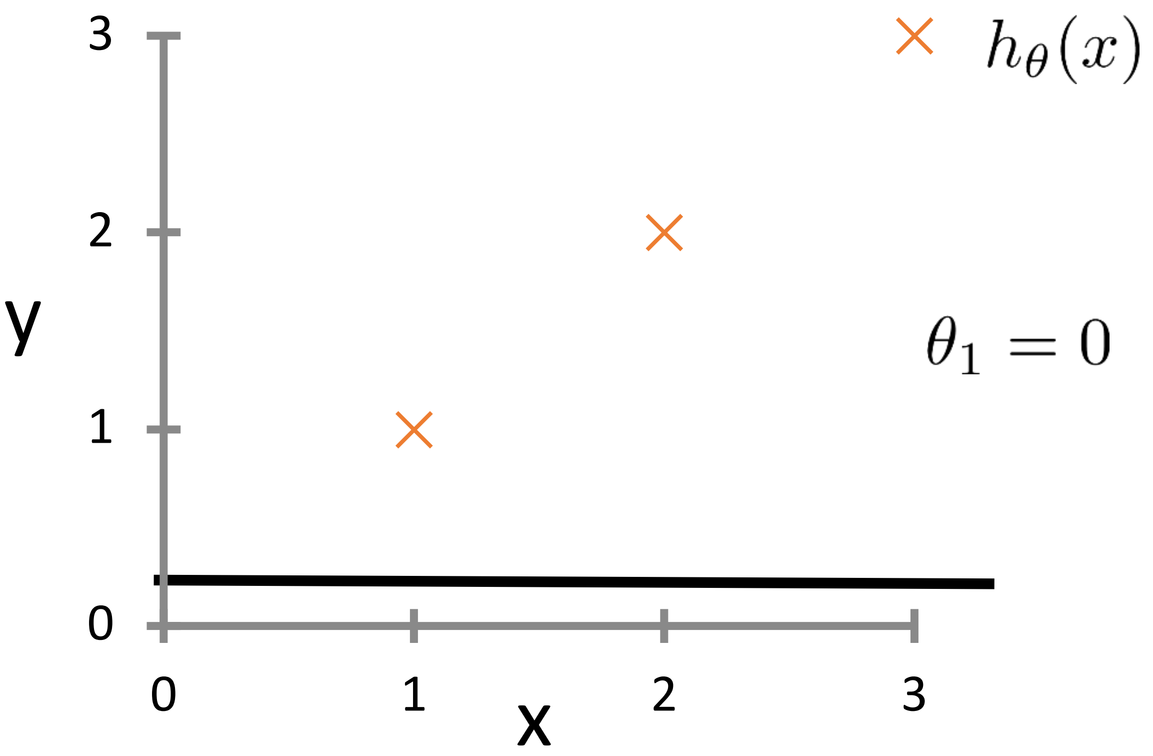
$$J(\theta_1)$$

(function of the parameter θ_1)



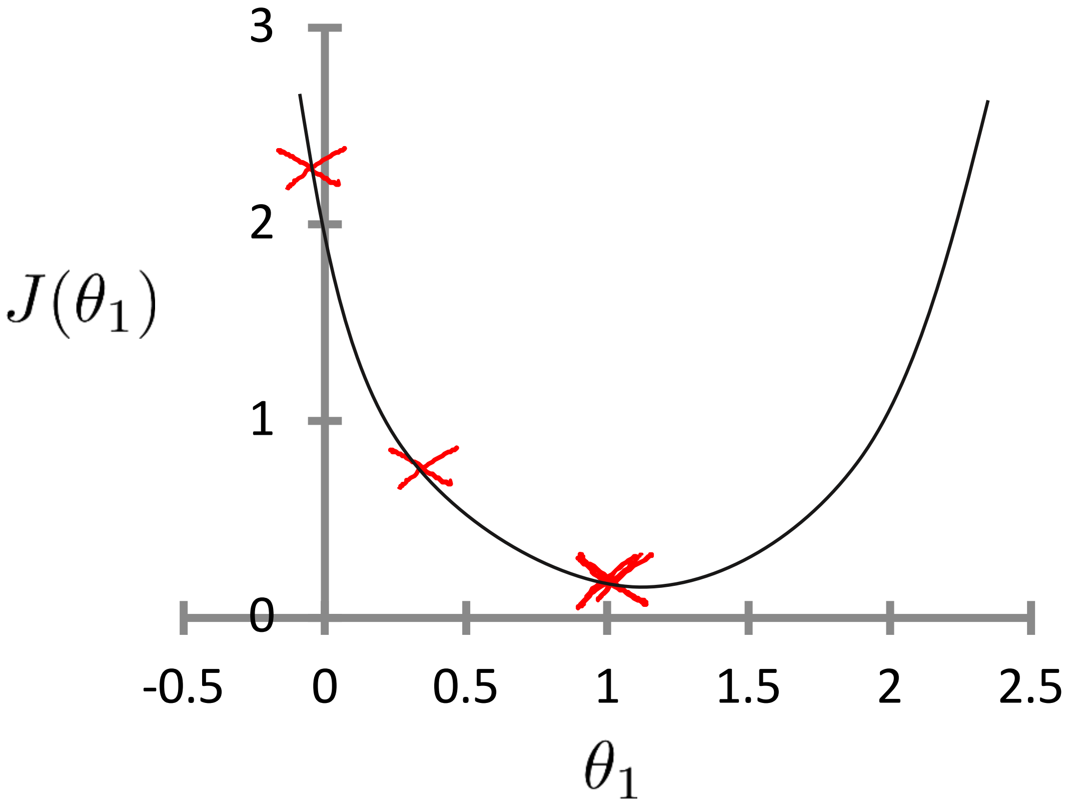
$h_\theta(x)$

(for fixed θ_1, this is a function of x)



$J(\theta_1)$

(function of the parameter θ_1)



Bringing back the intercept parameter

Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$

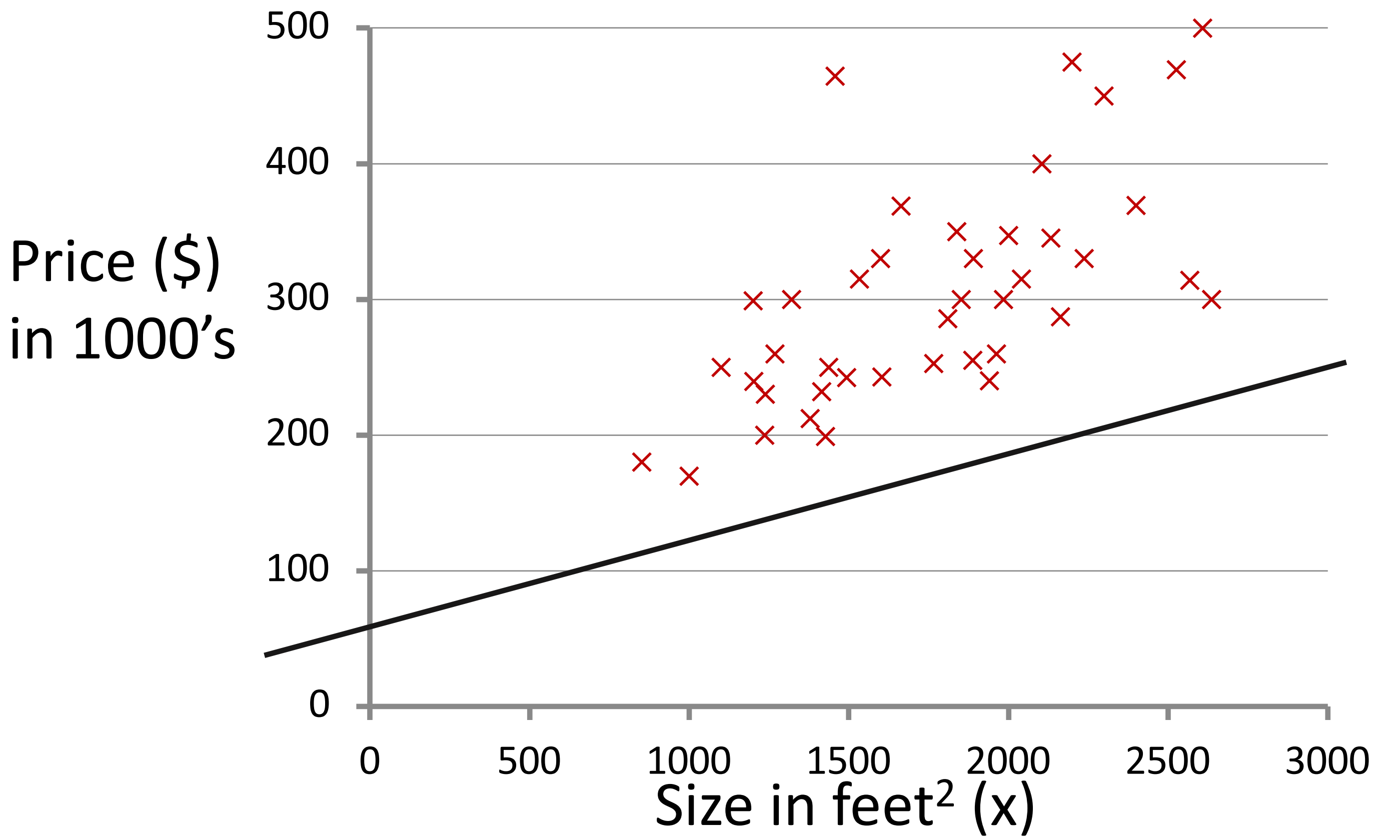
Parameters: θ_0, θ_1

Cost Function: $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$

Goal: $\underset{\theta_0, \theta_1}{\text{minimize}}\; J(\theta_0, \theta_1)$

$$h_\theta(x)$$

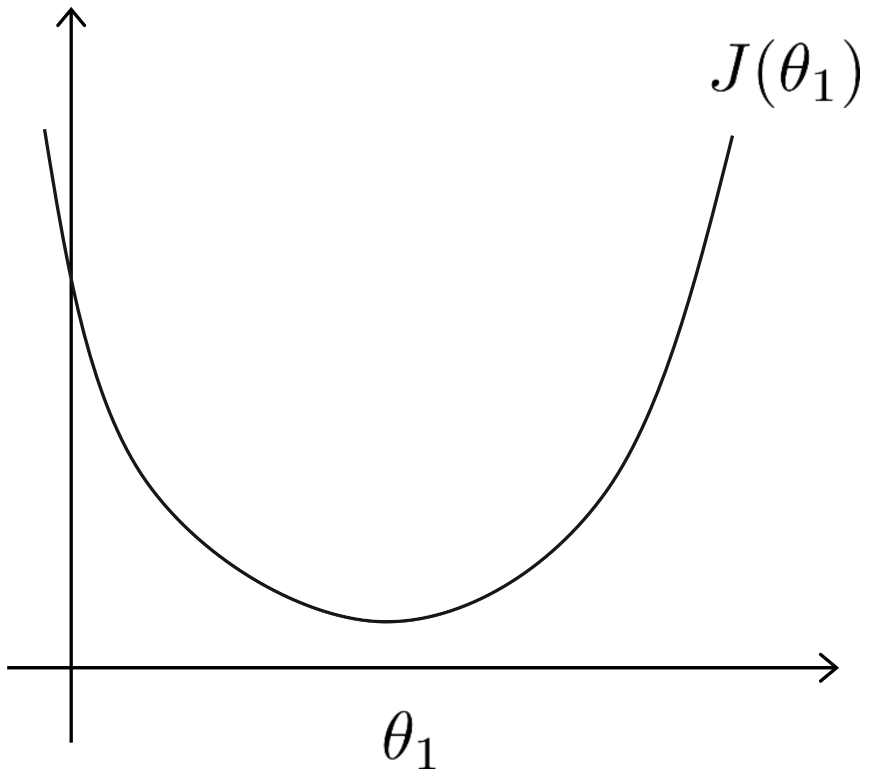
(for fixed θ_0, θ_1, this is a function of x)

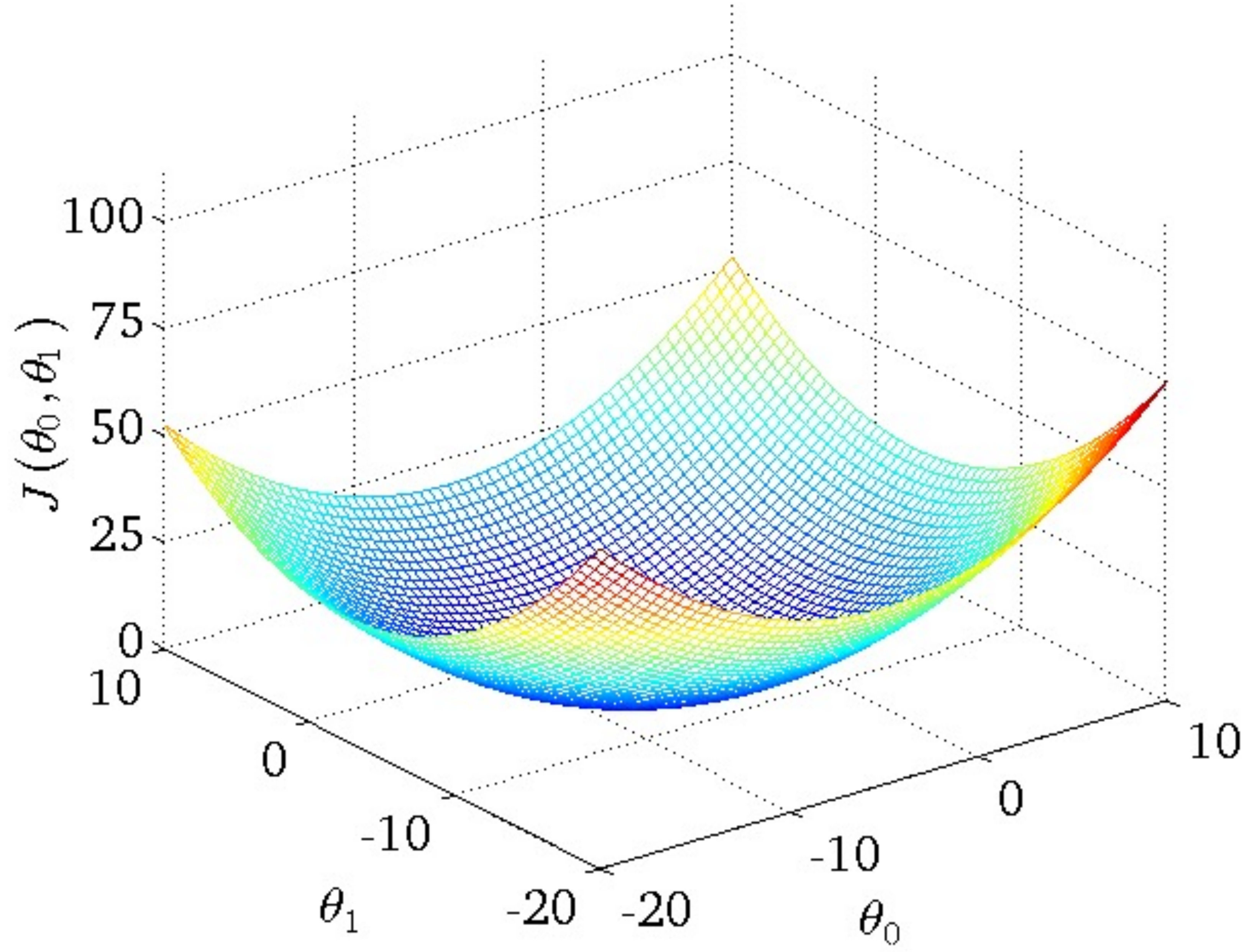


$$h_\theta(x) = 50 + 0.06x$$

$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)

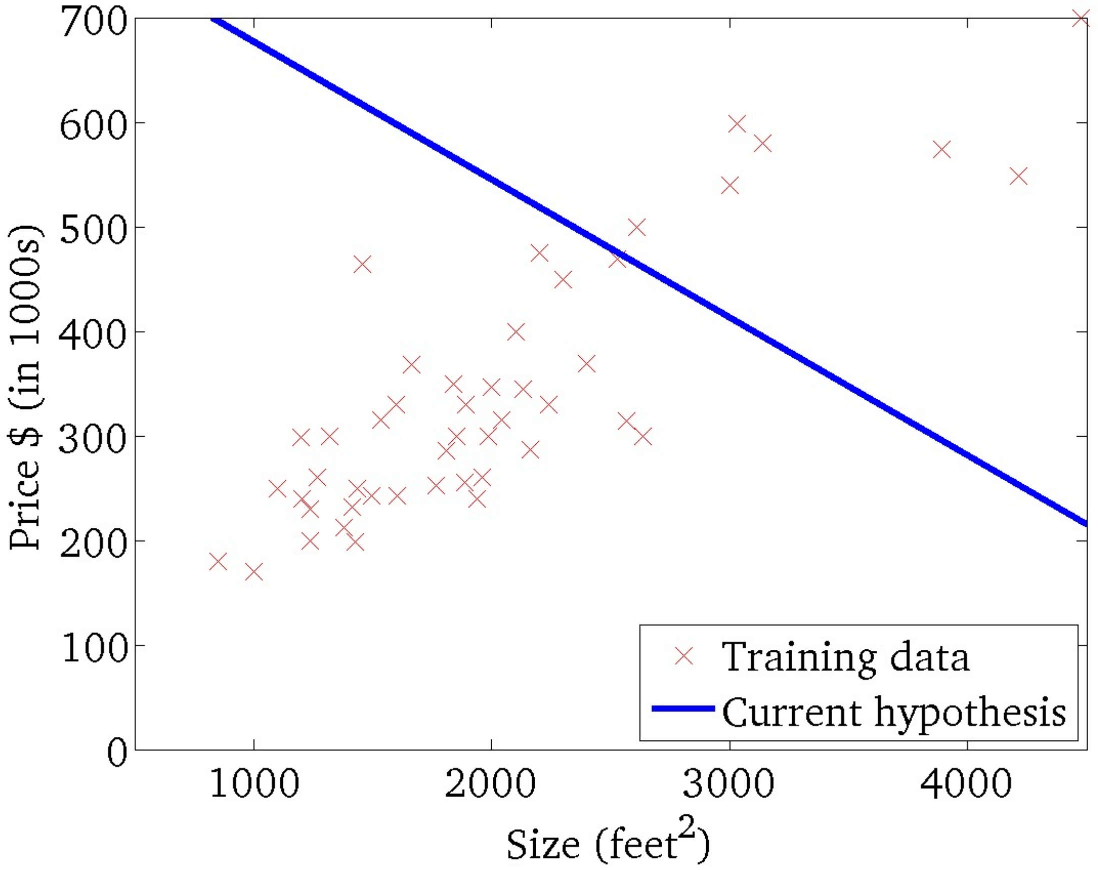




$J(\theta_0, \theta_1)$
100
75
50
25
0
10
0
-10
-20
θ_1
-20
-10
0
10
θ_0

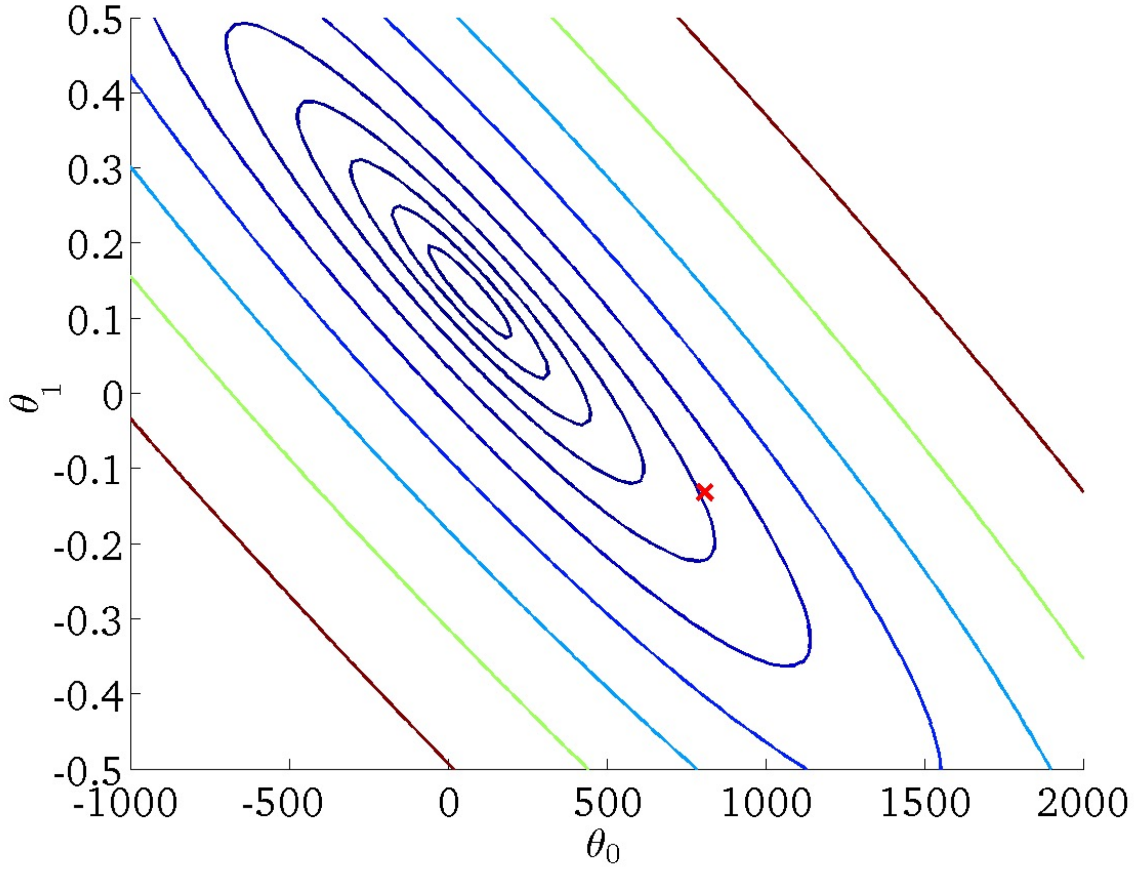
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



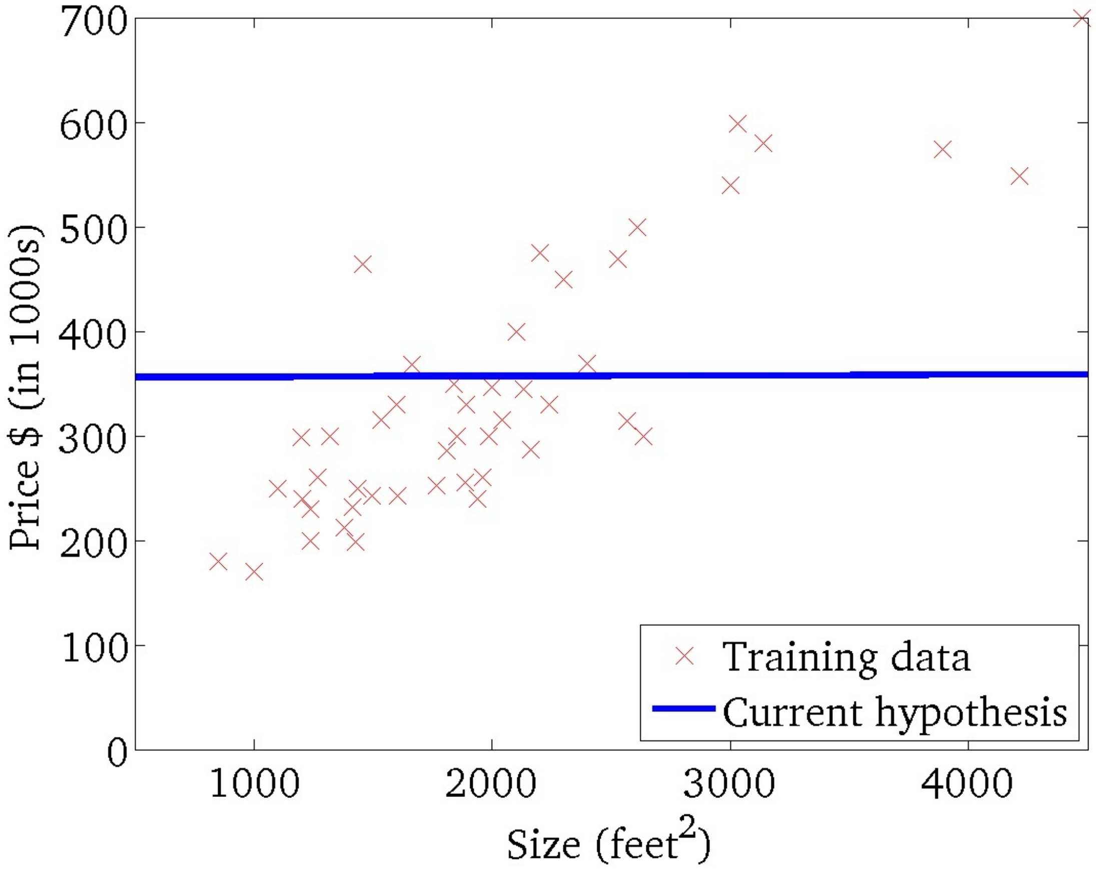
$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



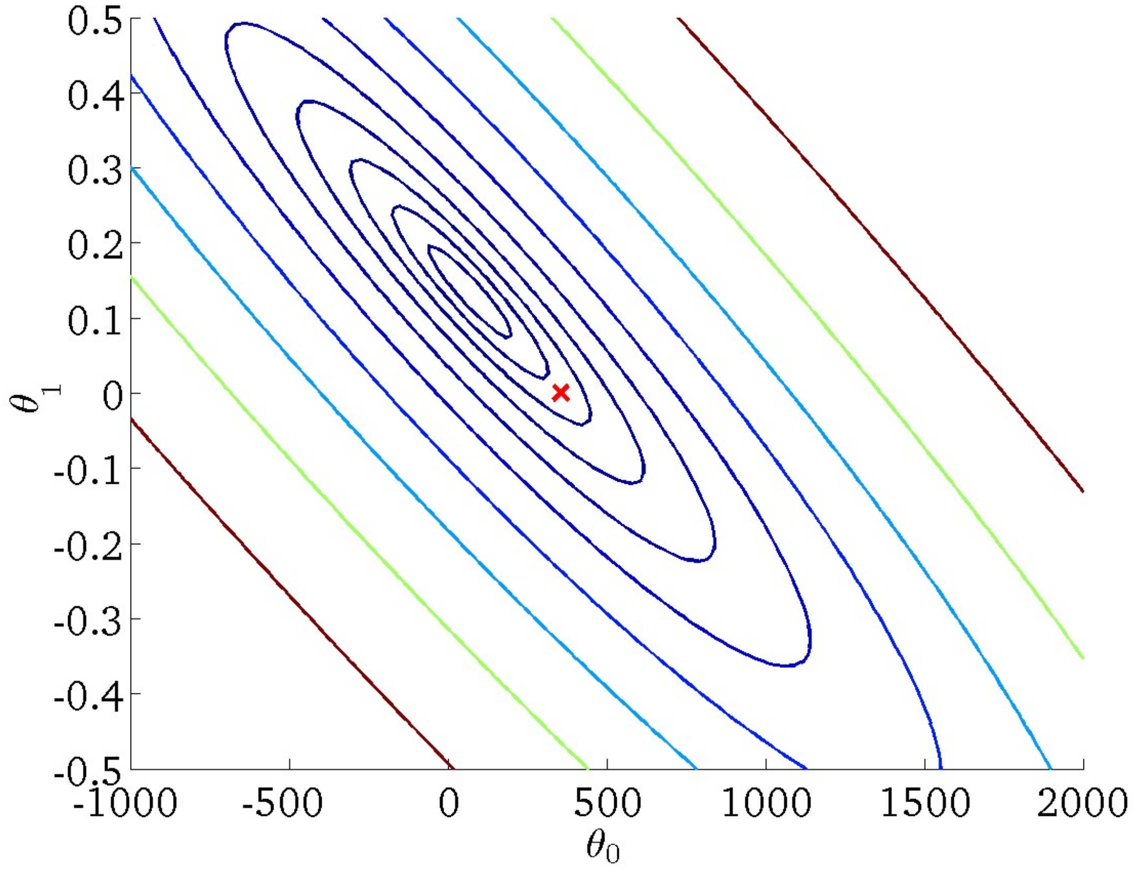
$$h_\theta(x)$$

(for fixed θ_0, θ_1, this is a function of x)



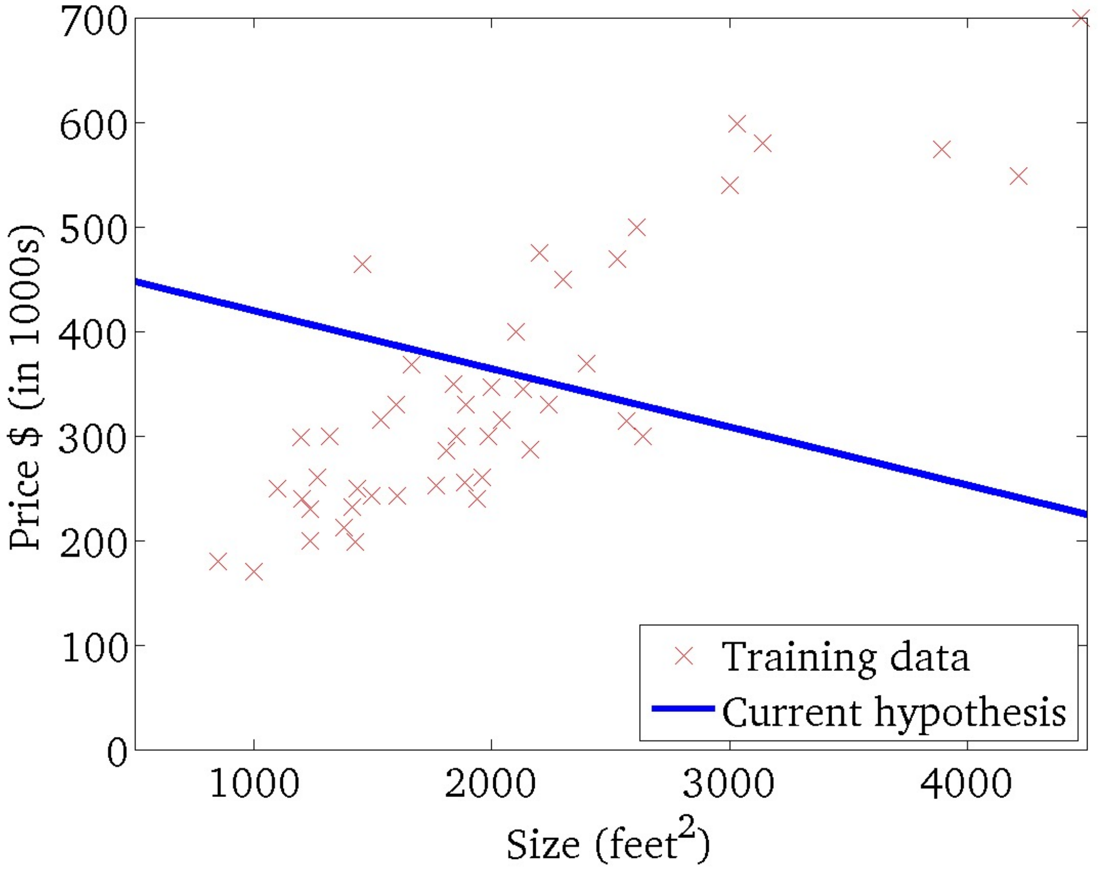
$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



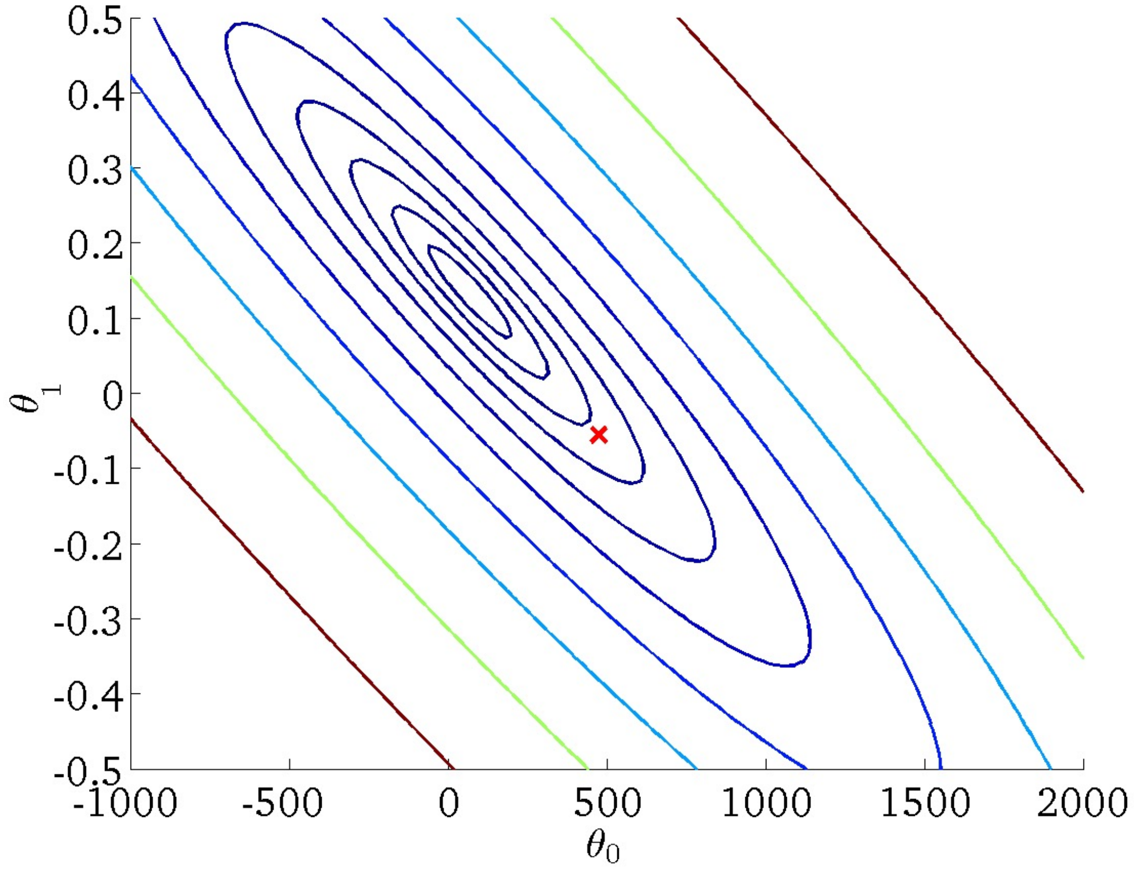
$$h_\theta(x)$$

(for fixed θ_0, θ_1, this is a function of x)



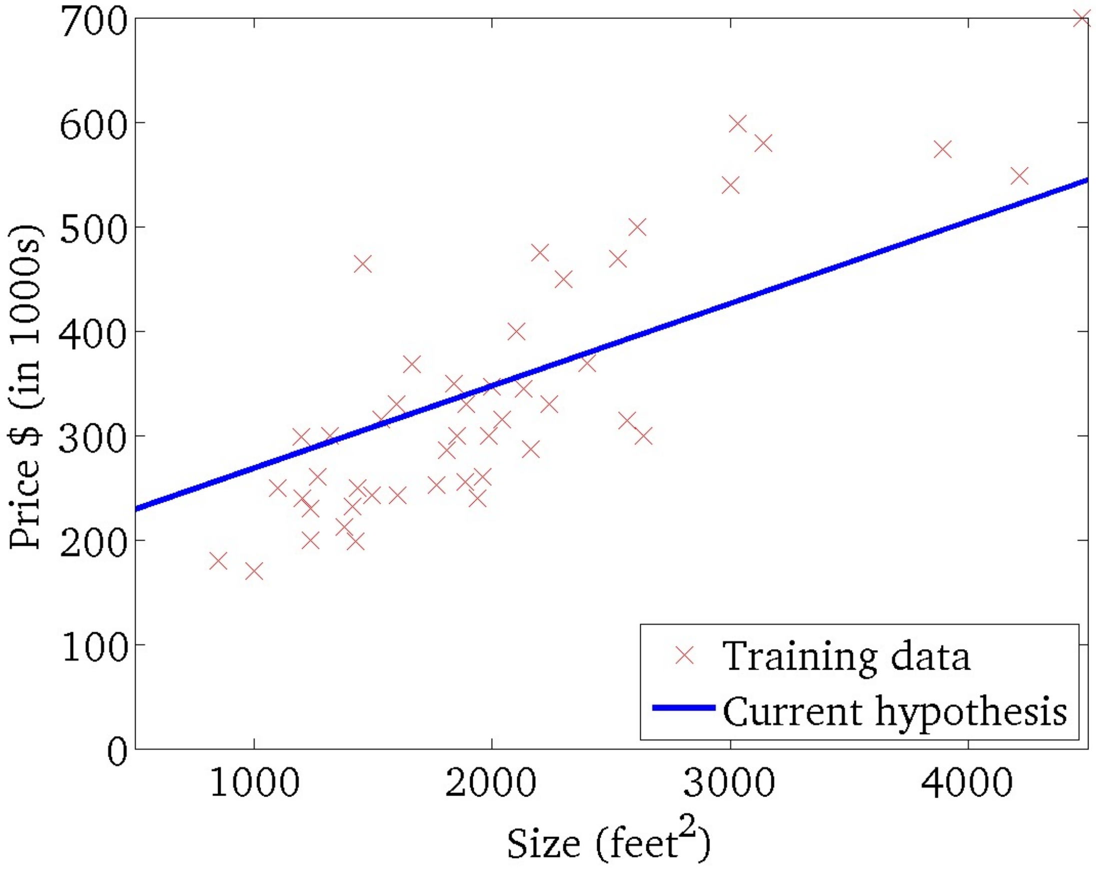
$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



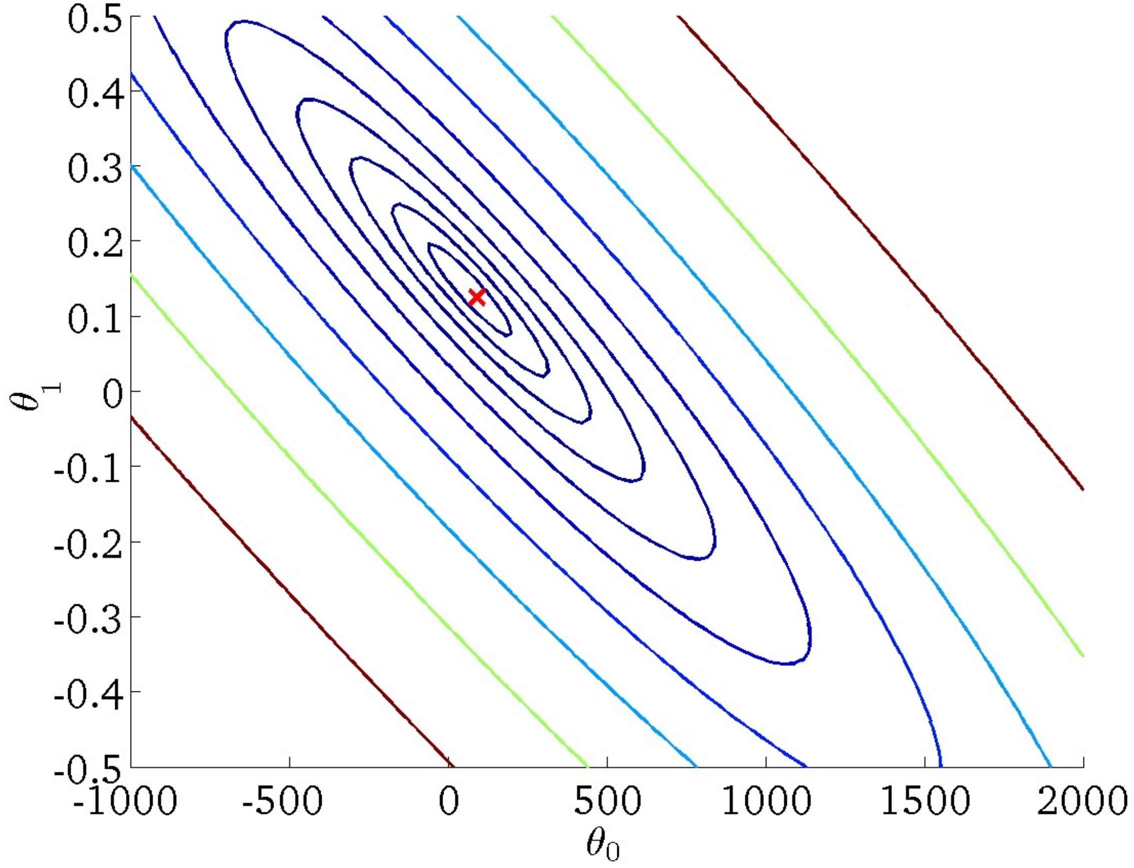
$$h_\theta(x)$$

(for fixed θ_0, θ_1, this is a function of x)



$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



How to find the best model parameters?

- For simple cases, one may find the optimal model parameters analytically (i.e. through calculus)
- How ever in most cases, this is not possible

Have some function $J(\theta_0, \theta_1)$

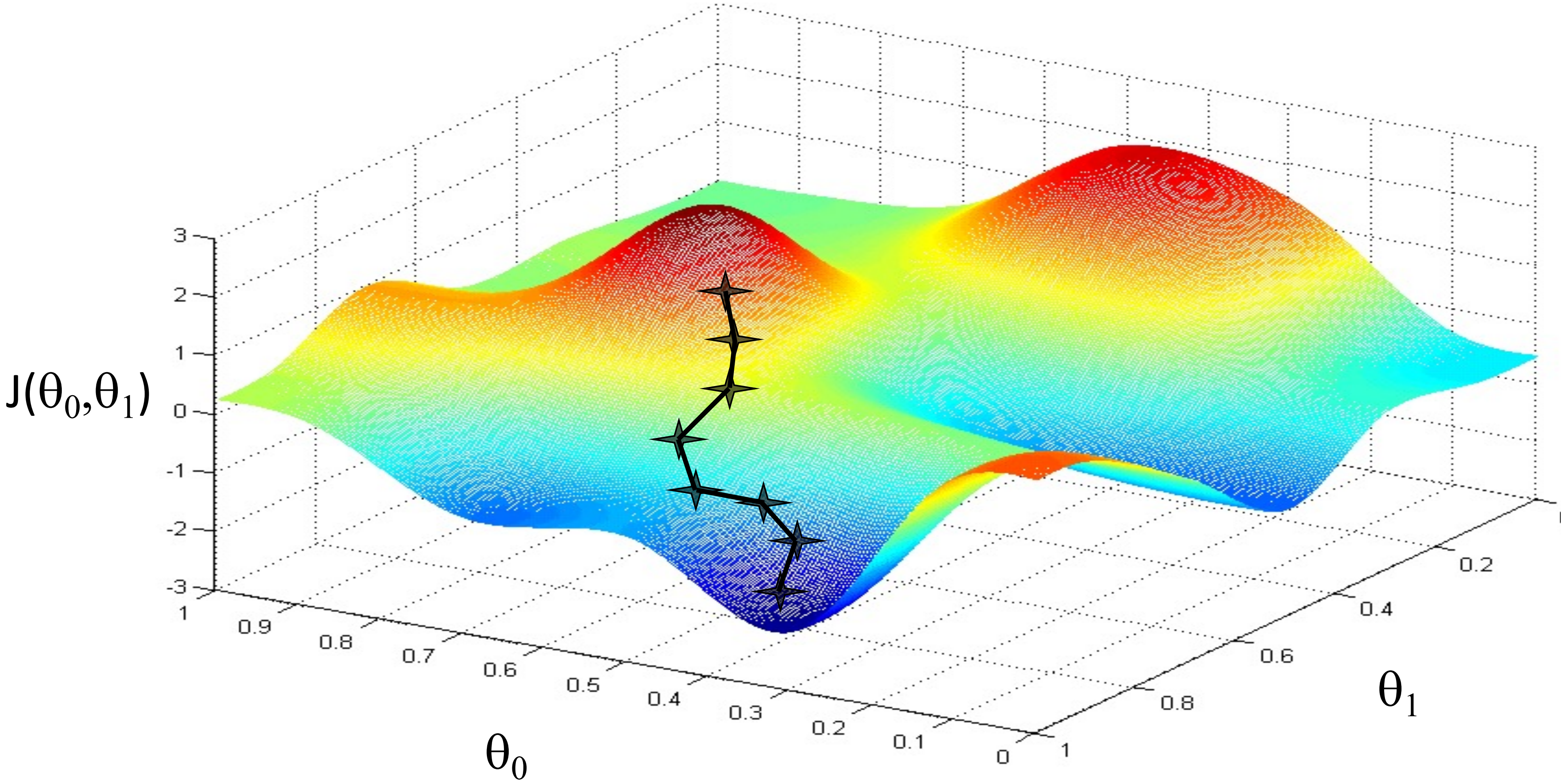
Want $\min_{\theta_0, \theta_1} J(\theta_0, \theta_1)$

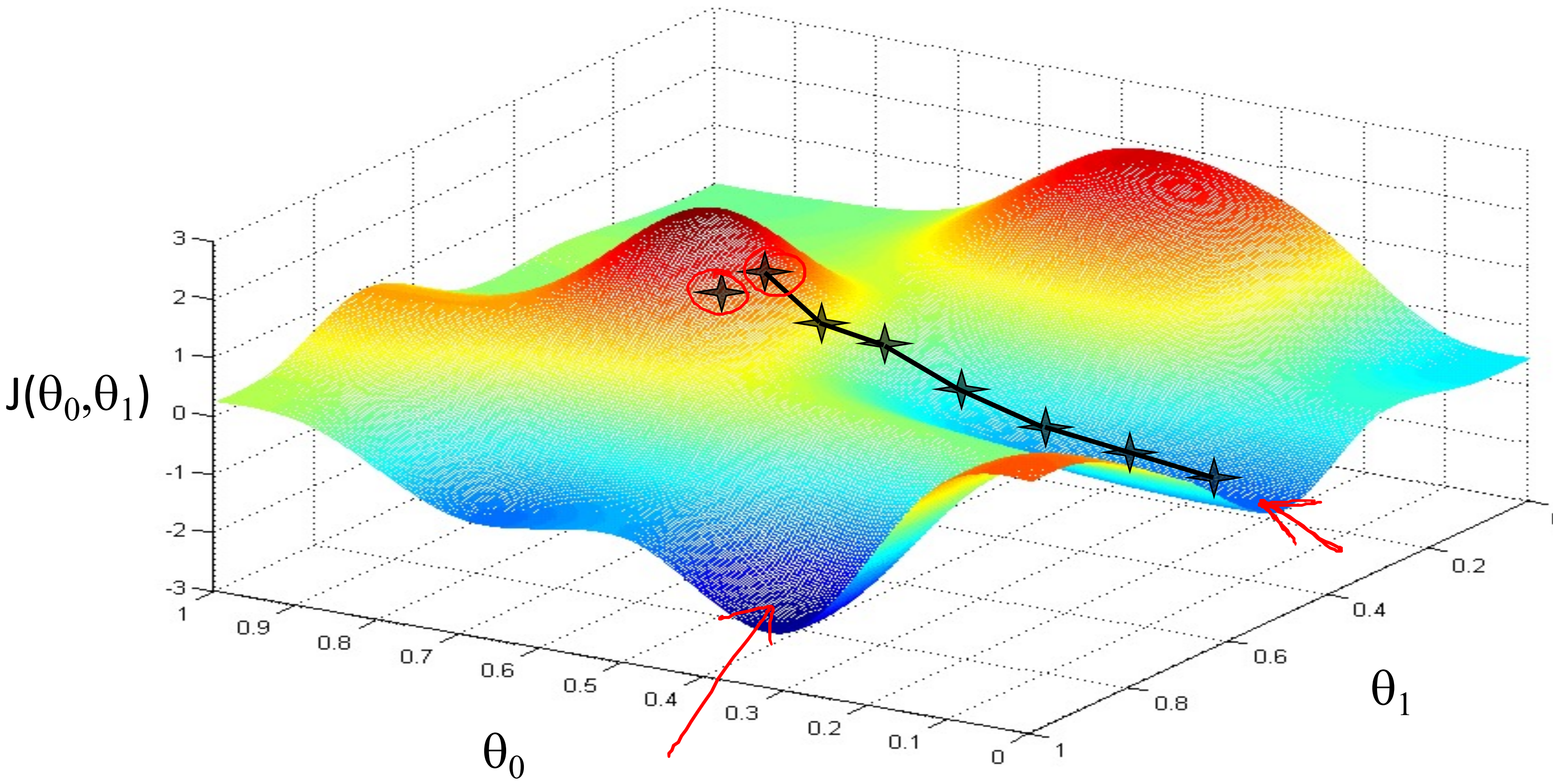
Outline:

- Start with some θ_0, θ_1
- Keep changing θ_0, θ_1 to reduce $J(\theta_0, \theta_1)$

  until we hopefully end up at a minimum

Introducing gradient decent





J(θ0,θ1)
θ0
θ1
3
2
1
0
-1
-2
-3
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1

Gradient descent algorithm

$$\text{repeat until convergence } \{$$
$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \qquad (\text{for } j = 0 \text{ and } j = 1)$$
$$\}$$

Correct: Simultaneous update

$$\text{temp0} := \theta_0 - \alpha \tfrac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$
$$\text{temp1} := \theta_1 - \alpha \tfrac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$
$$\theta_0 := \text{temp0}$$
$$\theta_1 := \text{temp1}$$

Incorrect:

$$\text{temp0} := \theta_0 - \alpha \tfrac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$
$$\theta_0 := \text{temp0}$$
$$\text{temp1} := \theta_1 - \alpha \tfrac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$
$$\theta_1 := \text{temp1}$$