BOSTON UNIVERSITY

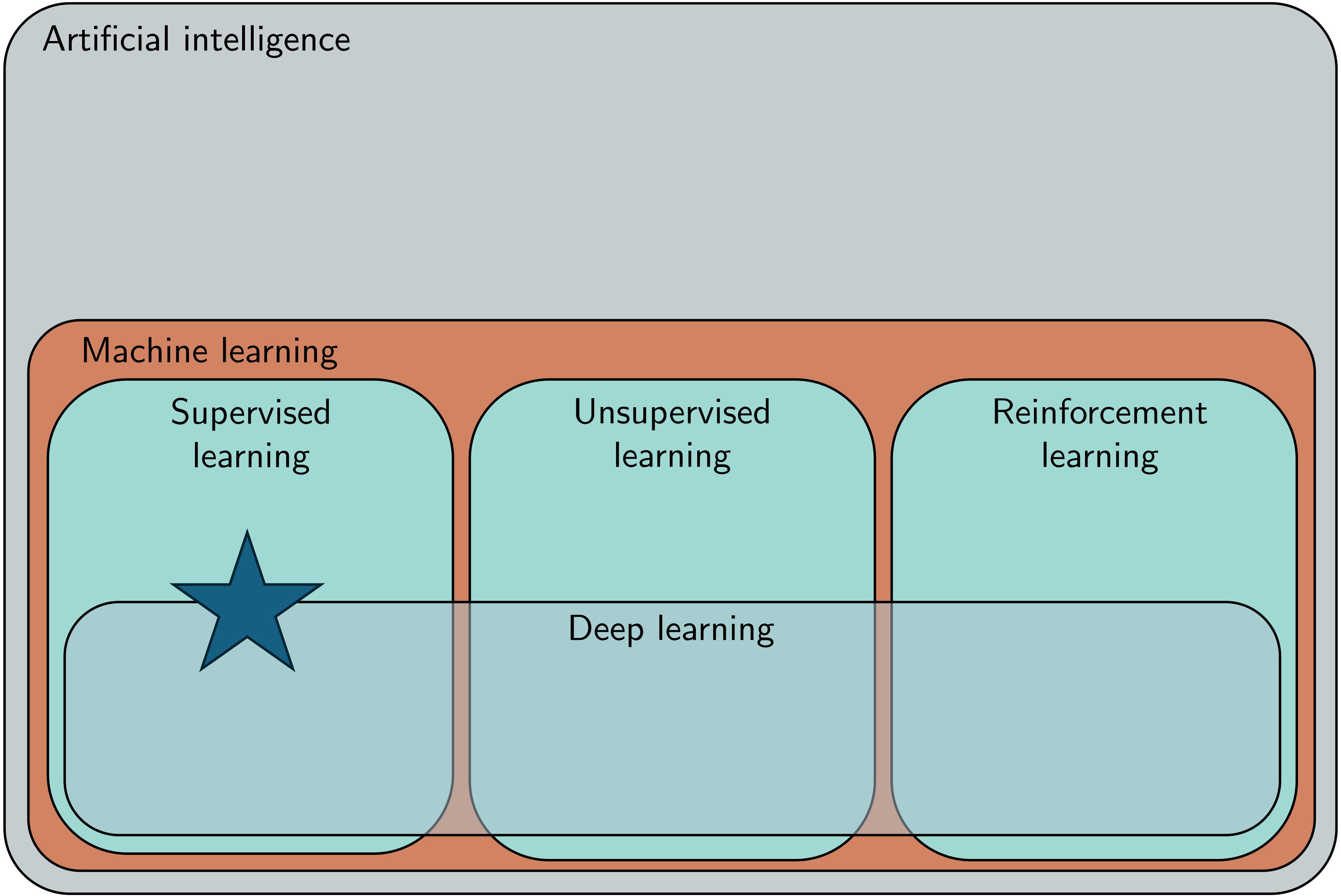
Deep Learning for Data Science
DS 542

https://dl4ds.github.io/sp2026/

Supervised Learning

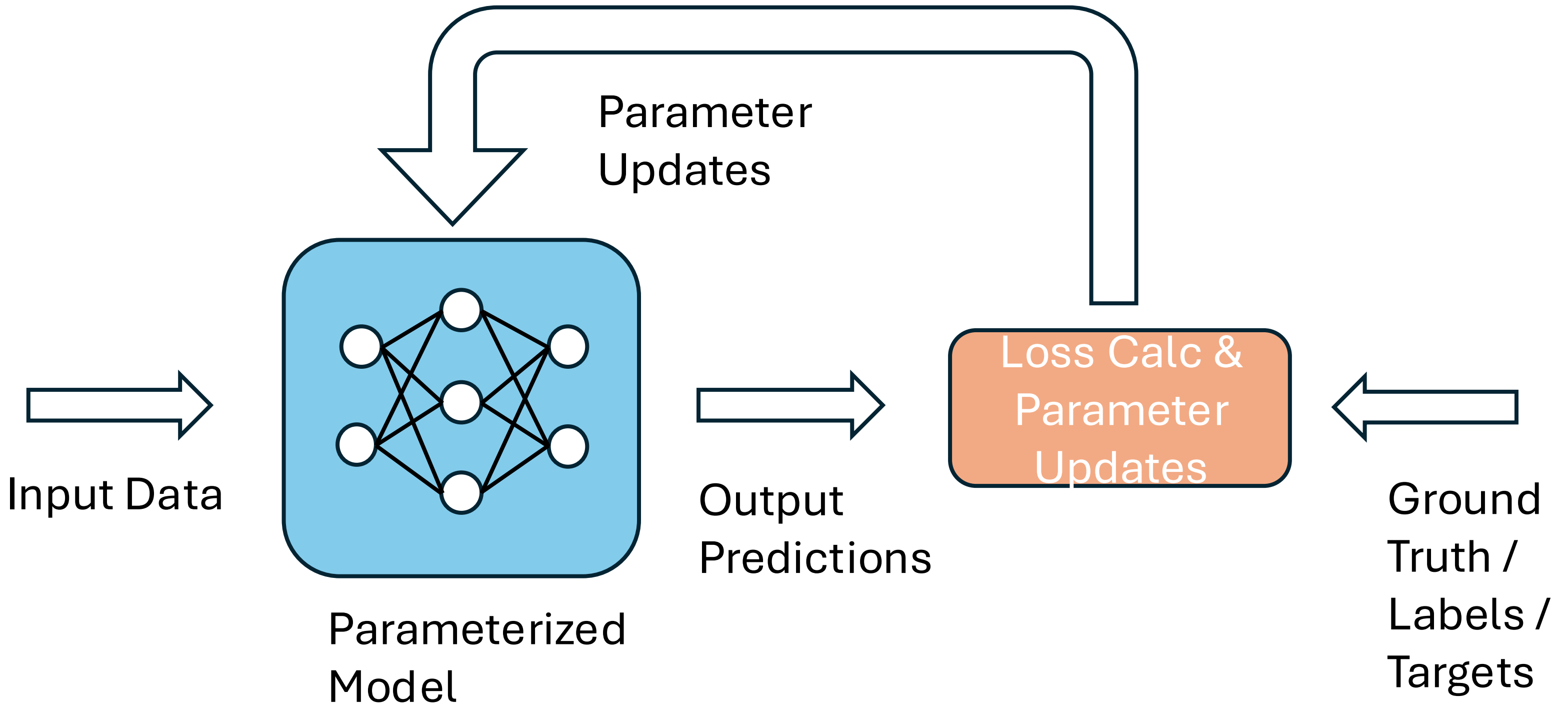
Supervised learning

- Examples
- Terminology
- Notation
  - Model
  - Loss function
  - Training
  - Testing
- 1D Linear regression example
  - Model
  - Loss function
  - Training
  - Testing

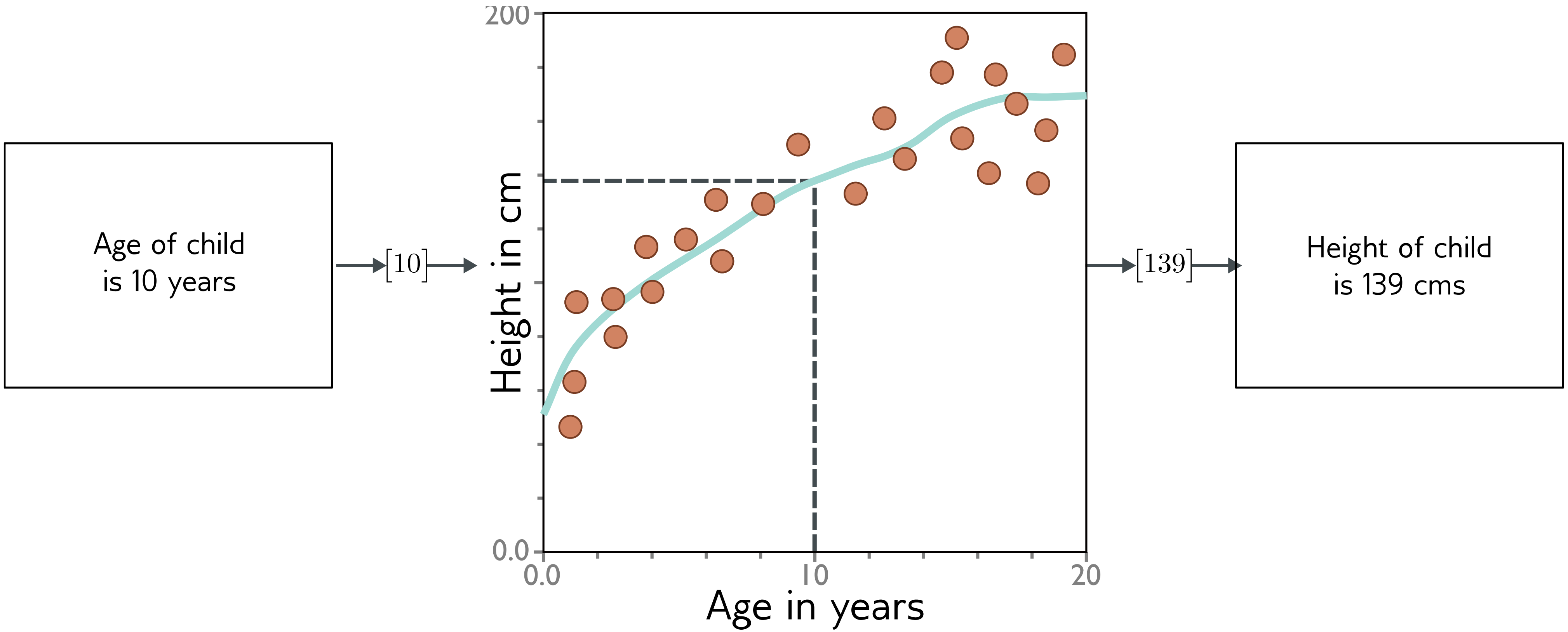

Artificial intelligence
Machine learning
Supervised learning
Unsupervised learning
Reinforcement learning
Deep learning

Supervised learning

- Define a mapping from input to output
- Learn this mapping from paired input/output data examples

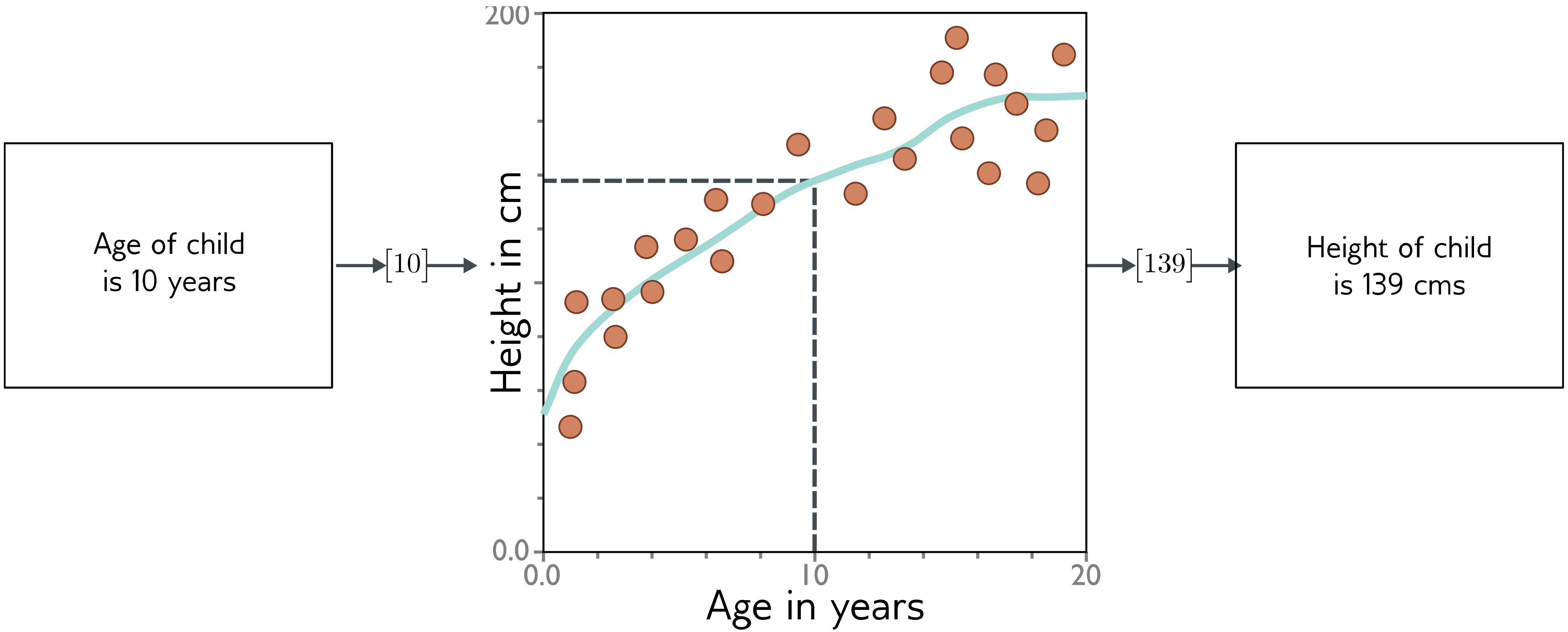


What is a supervised learning model?



- An equation relating input (age) to output (height)
- Search through family of possible equations to find one that fits training data well

What is a supervised learning model?



- Deep neural networks are just a very flexible family of equations
- Fitting deep neural networks = “Deep Learning”

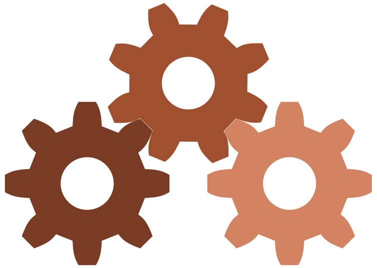
Prediction Types

- Regression
  - Prediction a continuous valued output

- Classification
  - Assigning input to one of a finite number of classes or categories
  - Two classes are a special case

Can be univariate (one output) or multivariate (more than one output)

Regression

Real world input	Model input	Model	Model output	Real world output
6000 square feet, 4 bedrooms, previously sold for $235K in 2005, 1 parking spot.	$\begin{bmatrix} 6000 \\ 4 \\ 235 \\ 2005 \\ 1 \end{bmatrix}$	 Supervised learning model	$[340]$	Predicted price is $340k

- Univariate regression problem (one output, real value)
- Fully connected network

Graph regression

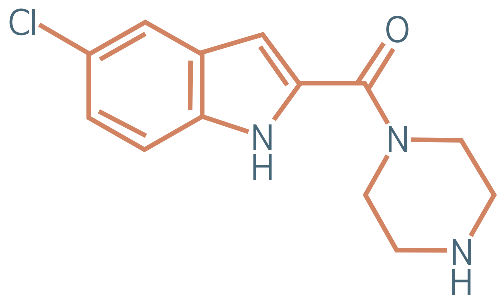
Real world input

Model input

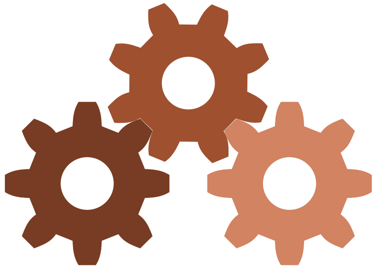
Model

Model output

Real world output



$$\begin{bmatrix} 1 \\ 0 \\ 1 \\ \vdots \\ 17 \\ 1 \\ 1 \\ \vdots \end{bmatrix}$$



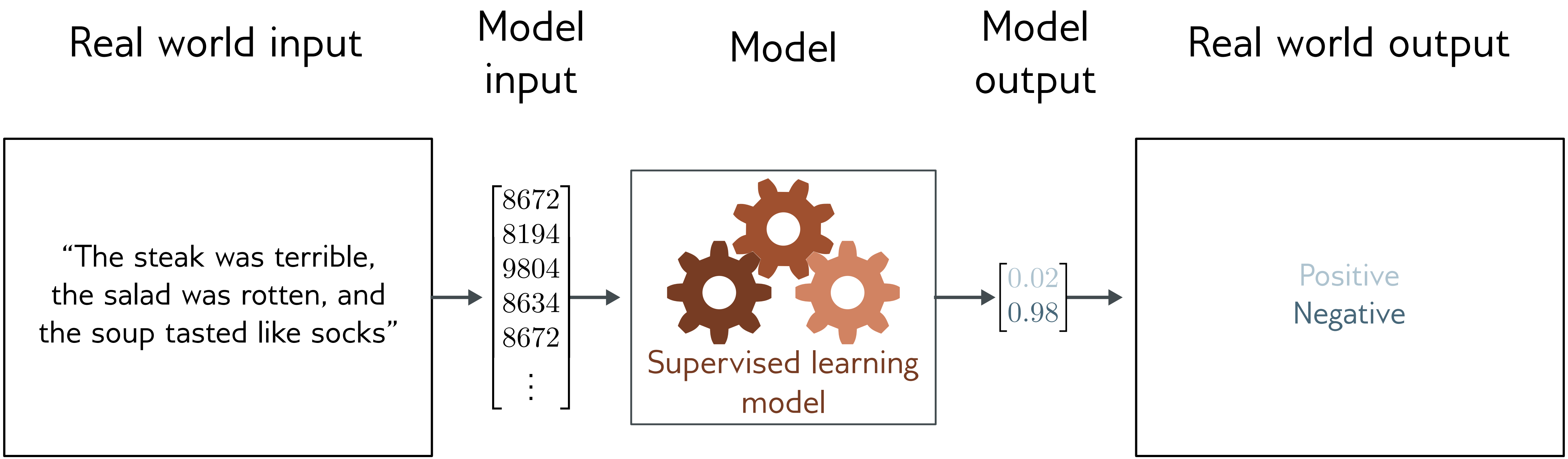
Supervised learning model

$$\begin{bmatrix} -12.9 \\ 56.4 \end{bmatrix}$$

Freezing point is -12.9°C
Boiling point is 56.4°C

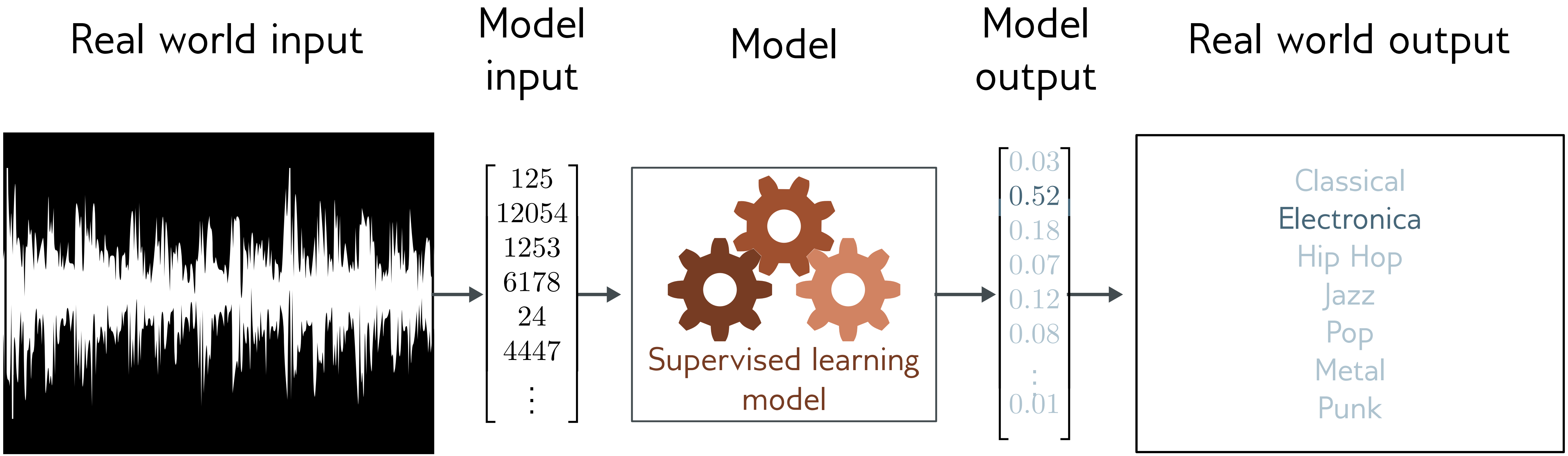
- Multivariate regression problem (>1 output, real value)
- Graph neural network

Text classification



- Binary classification problem (two discrete classes)
- Transformer network

Music genre classification

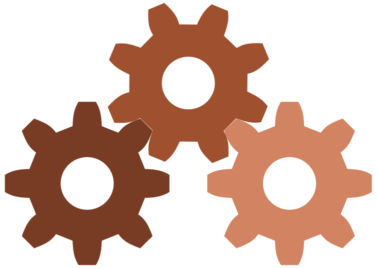


- Multiclass classification problem (discrete classes, >2 possible values)
- Recurrent neural network (RNN)

Image classification

Real world input | Model input | Model | Model output | Real world output

$$\begin{bmatrix} {\color{darkred}124} \\ {\color{green}140} \\ {\color{blue}156} \\ {\color{darkred}128} \\ {\color{green}142} \\ {\color{blue}157} \\ \vdots \end{bmatrix}$$



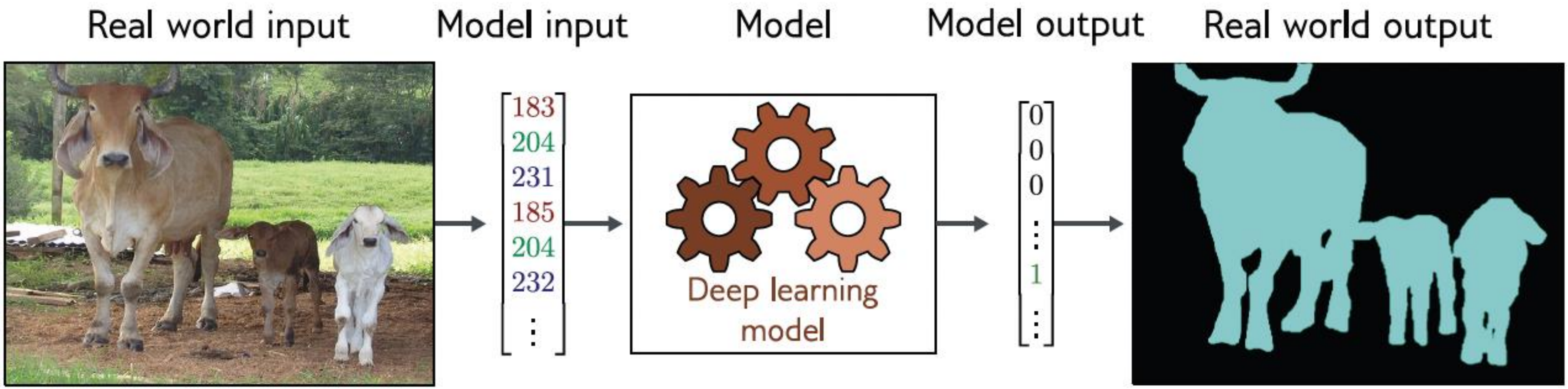
Supervised learning model

$$\begin{bmatrix} 0.00 \\ 0.00 \\ 0.01 \\ 0.89 \\ 0.05 \\ 0.00 \\ \vdots \\ 0.01 \end{bmatrix}$$

Aardvark
Apple
Bee
Bicycle
Bridge
Clown
⋮

- Multiclass classification problem (discrete classes, >2 possible classes)
- Convolutional network

Image segmentation



- Multivariate binary classification problem (many outputs, two discrete classes)
- Convolutional encoder-decoder network

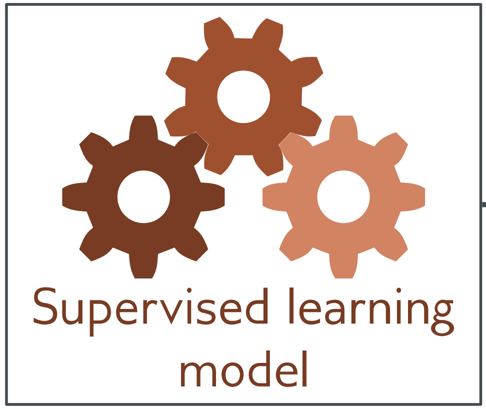
Depth estimation

Real world input

Model input

$$\begin{bmatrix}255\\254\\255\\254\\254\\255\\\vdots\end{bmatrix}$$

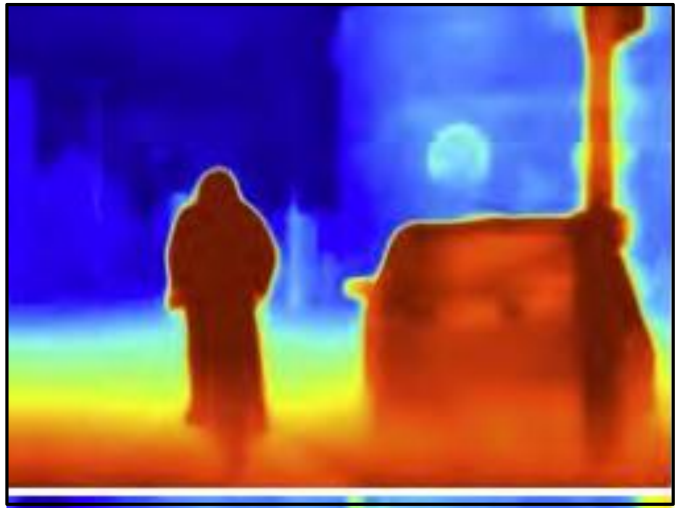
Model



Model output

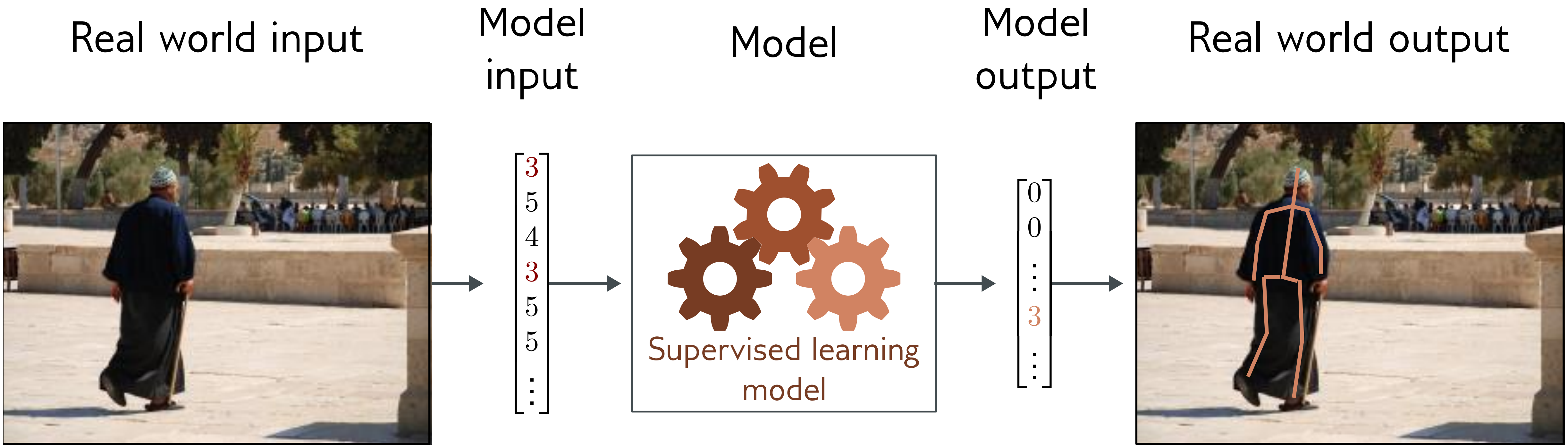
$$\begin{bmatrix}0.001\\0.002\\\vdots\\0.314\\0.310\\\vdots\end{bmatrix}$$

Real world output



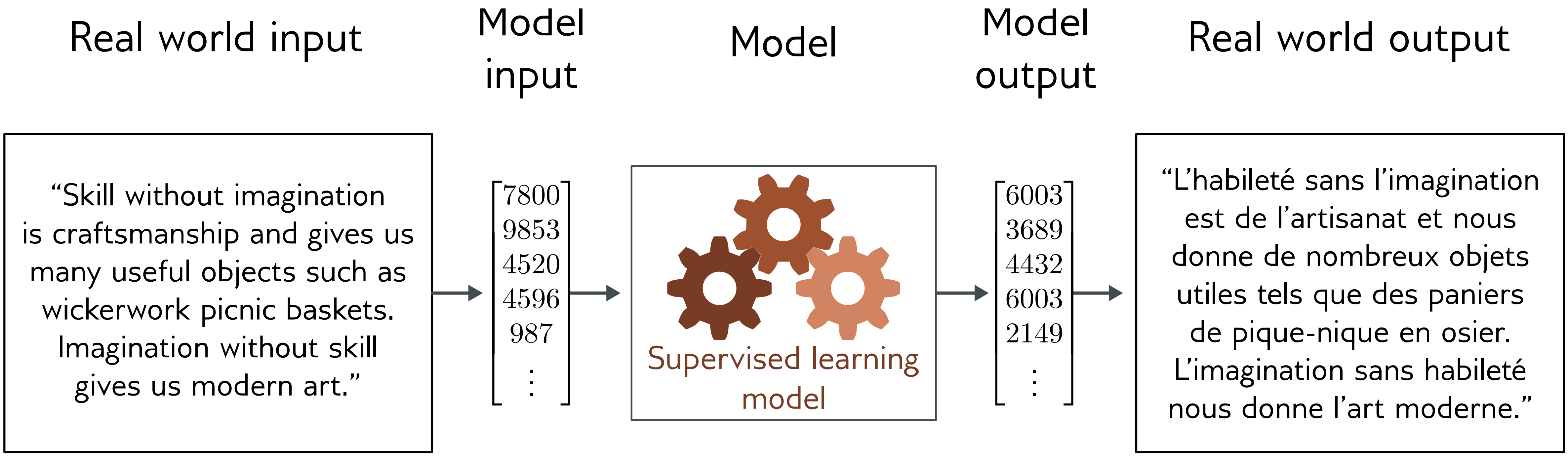
- Multivariate regression problem (many outputs, continuous)
- Convolutional encoder-decoder network

Pose estimation



- Multivariate regression problem (many outputs, continuous)
- Convolutional encoder-decoder network

Translation

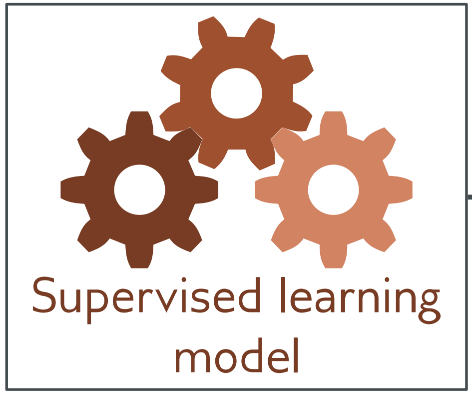


- Encoder-Decoder Transformer Networks

Image captioning

Real world input | Model input | Model | Model output | Real world output

$$\begin{bmatrix} 183 \\ 204 \\ 231 \\ 185 \\ 204 \\ 232 \\ \vdots \end{bmatrix}$$

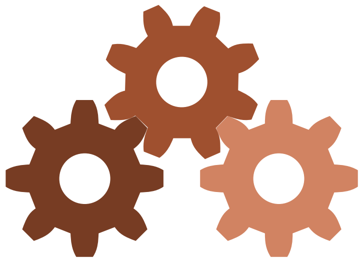


$$\begin{bmatrix} 1 \\ 5593 \\ 7532 \\ 7924 \\ 1 \\ \vdots \end{bmatrix}$$

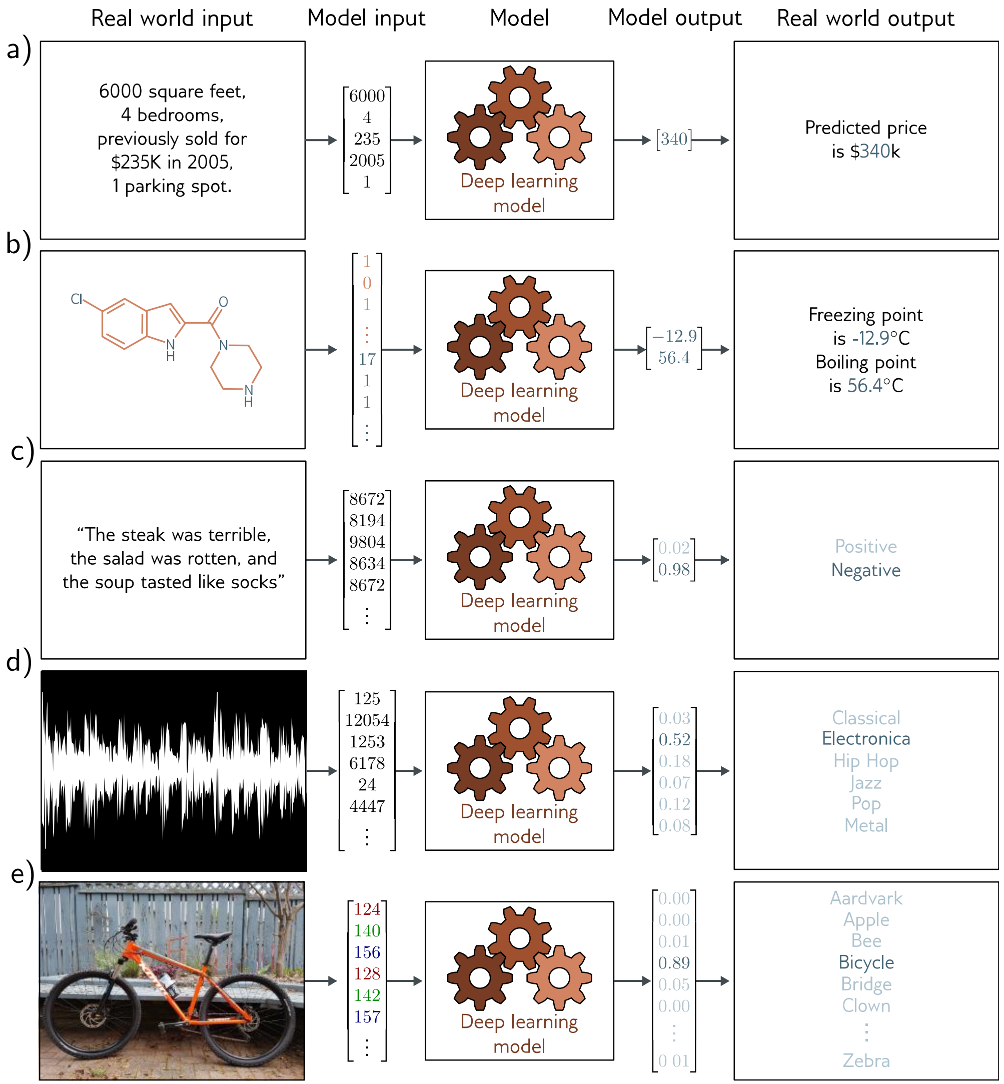
"A Kazakh man on a horse holding a bird of prey"

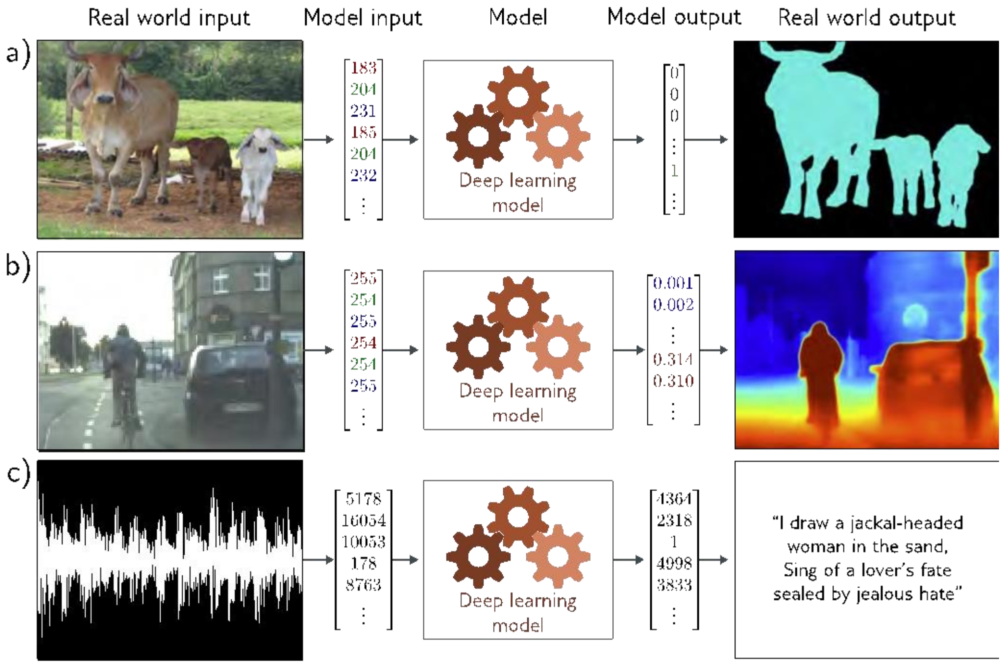
- E.g. CNN-RNN, LSTM, Transformers

Image generation from text

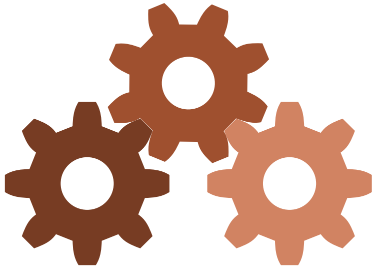
Real world input	Model input	Model	Model output	Real world output
"Teddy bears mixing sparkling chemicals as mad scientists, in a steampunk style."	$\begin{bmatrix} 8300 \\ 532 \\ 7676 \\ 7898 \\ 883 \\ \vdots \end{bmatrix}$	 Supervised learning model	$\begin{bmatrix} 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 1 \\ \vdots \end{bmatrix}$	

Supervised Learning Classification and Regression Applications





Regression

Real world input	Model input	Model	Model output	Real world output
6000 square feet, 4 bedrooms, previously sold for $235K in 2005, 1 parking spot.	$\begin{bmatrix} 6000 \\ 4 \\ 235 \\ 2005 \\ 1 \end{bmatrix}$	 Supervised learning model	$[340]$	Predicted price is $340k

- Univariate regression problem (one output, real value)

Any Questions?

Supervised learning

- Examples
- Terminology
- Notation
    - Model
    - Loss function
    - Training
    - Testing
- 1D Linear regression example
    - Model
    - Loss function
    - Training
    - Testing

Supervised learning terminology

- Supervised learning model = mapping from one or more inputs to one or more outputs
- Model is a family of equations → "inductive bias"
- Computing the outputs from the inputs → inference
- Model also includes parameters
- Parameters affect outcome of equation
- Training a model = finding parameters that predict outputs "well" from inputs for training and evaluation datasets of input/output pairs

Supervised learning

- Examples
- Terminology
- Notation
  - Model
  - Loss function
  - Training
  - Testing
- 1D Linear regression example
  - Model
  - Loss function
  - Training
  - Testing

Also Appendix A of the book.

Notation:

- Input:

$$\mathbf{x}$$

Variables always Roman letters

Normal lower case = scalar
Bold lower case = vector
Capital Bold = matrix

- Output:

$$\mathbf{y}$$

- Model:

$$\mathbf{y} = \mathbf{f}[\mathbf{x}]$$

Functions always square brackets

Normal lower case = returns scalar
Bold lower case = returns vector
Capital Bold = returns matrix

Notation example:

- Input:

$$\mathbf{x} = \begin{bmatrix} \text{age} \\ \text{mileage} \end{bmatrix}$$

Vector:
Structured or
tabular data

- Output:

$$y = \begin{bmatrix} \text{price} \end{bmatrix}$$

Scalar output

- Model:

$$y = \mathrm{f}[\mathbf{x}]$$

Scalar output
function
(with vector input)

Model

- Parameters:

$$\phi$$

Parameters always Greek letters

- Model :

$$\mathbf{y} = \mathbf{f}[\mathbf{x}, \phi]$$

Data Set and Loss function

- Training dataset of I pairs of input/output examples:

$$\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{I}$$

Data Set and Loss function

- Training dataset of I pairs of input/output examples:

$$\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{I}$$

- Loss function or cost function measures how bad model is:

$$L\Big[\phi, \underbrace{\mathrm{f}[\mathbf{x}, \phi]}_{\text{model}}, \underbrace{\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{I}}_{\text{train data}}\Big]$$

Data Set and Loss function

- Training dataset of I pairs of input/output examples:

$$\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{I}$$

- Loss function or cost function measures how bad model is:

$$L\Big[\underbrace{\phi, \mathrm{f}[\mathbf{x}, \phi]}_{\text{model}}, \underbrace{\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{I}}_{\text{train data}}\Big]$$

or for short:

$$L[\phi] \longleftarrow$$

Returns a scalar that is smaller when model maps inputs to outputs better

Training

- Loss function:

$$L\left[\phi\right]$$

Returns a scalar that is smaller when model maps inputs to outputs better

- Find the parameters that minimize the loss:

$$\hat{\phi} = \underset{\phi}{\operatorname{argmin}}\Big[\mathrm{L}\left[\phi\right]\Big]$$

Any Questions?

Supervised learning

- Examples
- Terminology
- Notation
  - Model
  - Loss function
  - Training
  - Testing
- 1D Linear regression example
  - Model
  - Loss function
  - Training
  - Testing

Example: 1D Linear regression model

- Model:

$$\begin{aligned} y &= \mathrm{f}[x, \boldsymbol{\phi}] \\ &= \phi_0 + \phi_1 x \end{aligned}$$

- Parameters

$$\boldsymbol{\phi} = \begin{bmatrix} \phi_0 \\ \phi_1 \end{bmatrix} \begin{matrix} \longleftarrow \text{y-offset} \\ \longleftarrow \text{slope} \end{matrix}$$

Example: 1D Linear regression model

- Model:

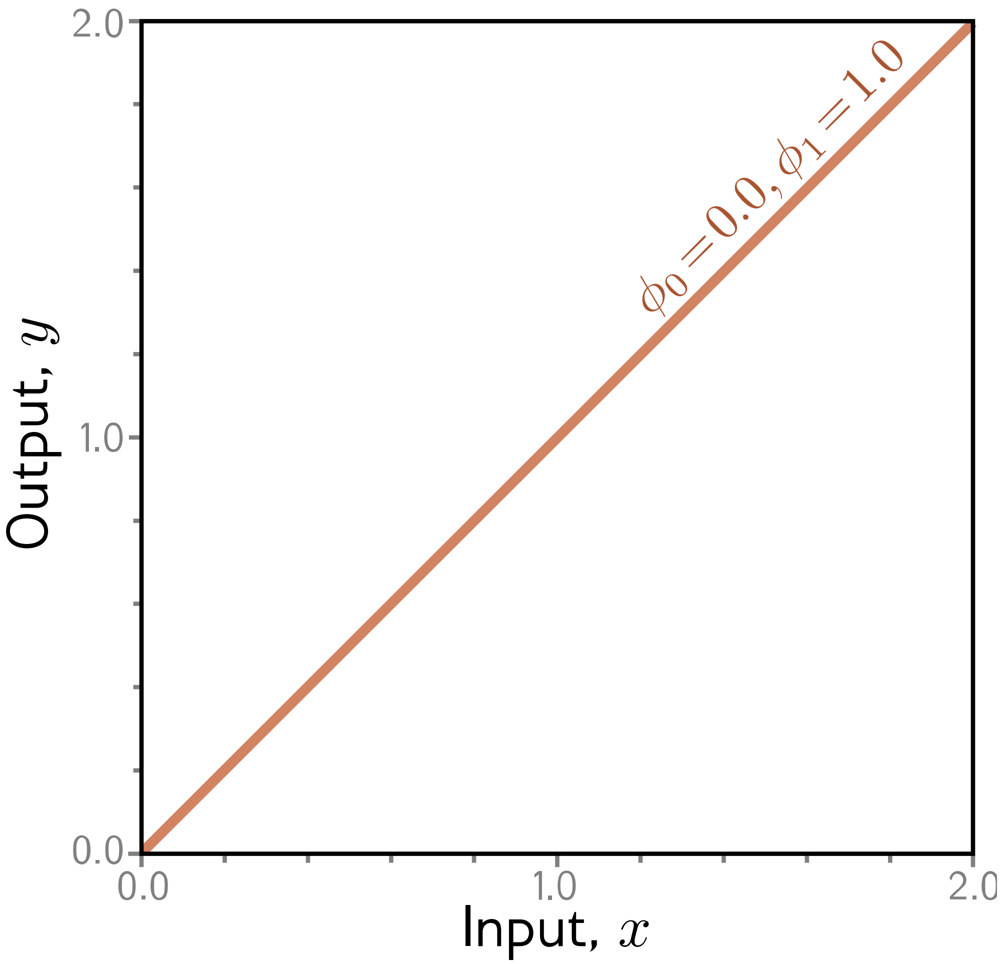
$$
\begin{aligned}
y &= \mathrm{f}[x, \boldsymbol{\phi}] \\
&= \phi_0 + \phi_1 x
\end{aligned}
$$

- Parameters

$$
\boldsymbol{\phi} = \begin{bmatrix} \phi_0 \\ \phi_1 \end{bmatrix}
$$

y-offset

slope



Example: 1D Linear regression model

- Model:

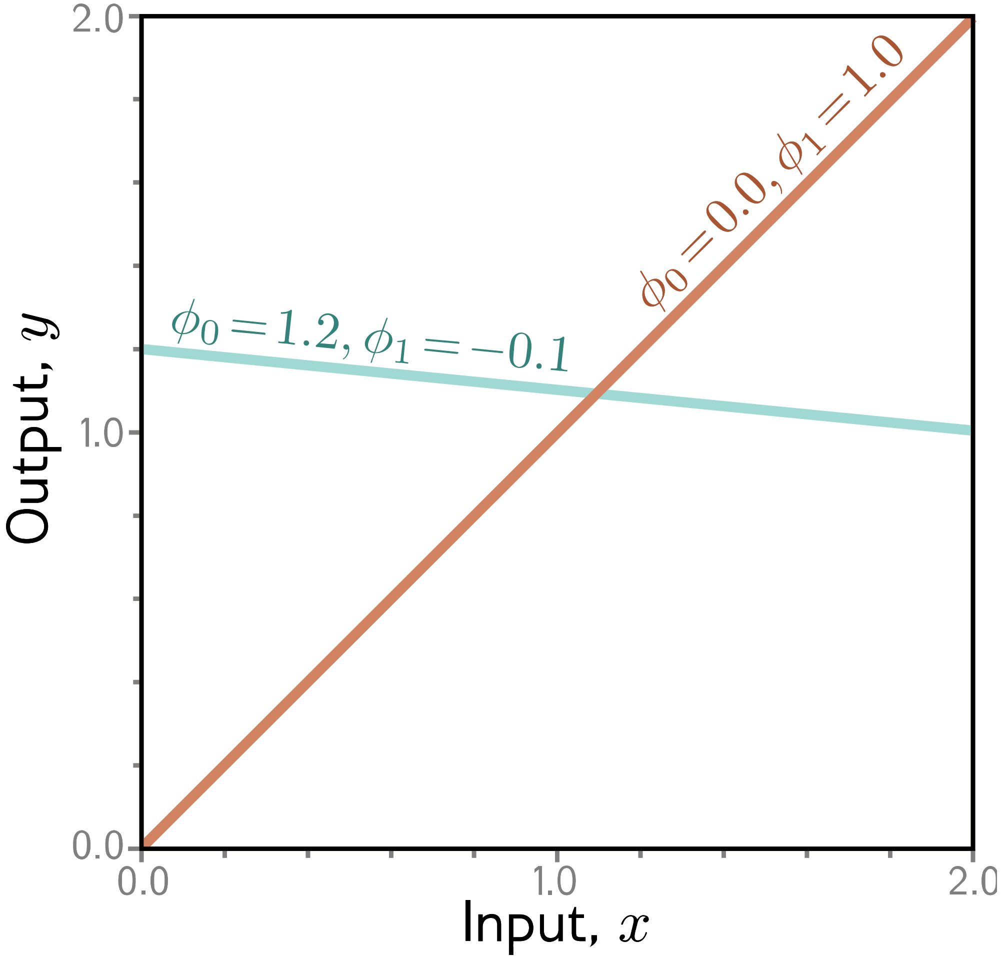
$$
\begin{aligned}
y &= \mathrm{f}[x, \boldsymbol{\phi}] \\
&= \phi_0 + \phi_1 x
\end{aligned}
$$

- Parameters

$$
\boldsymbol{\phi} = \begin{bmatrix} \phi_0 \\ \phi_1 \end{bmatrix}
$$

y-offset

slope



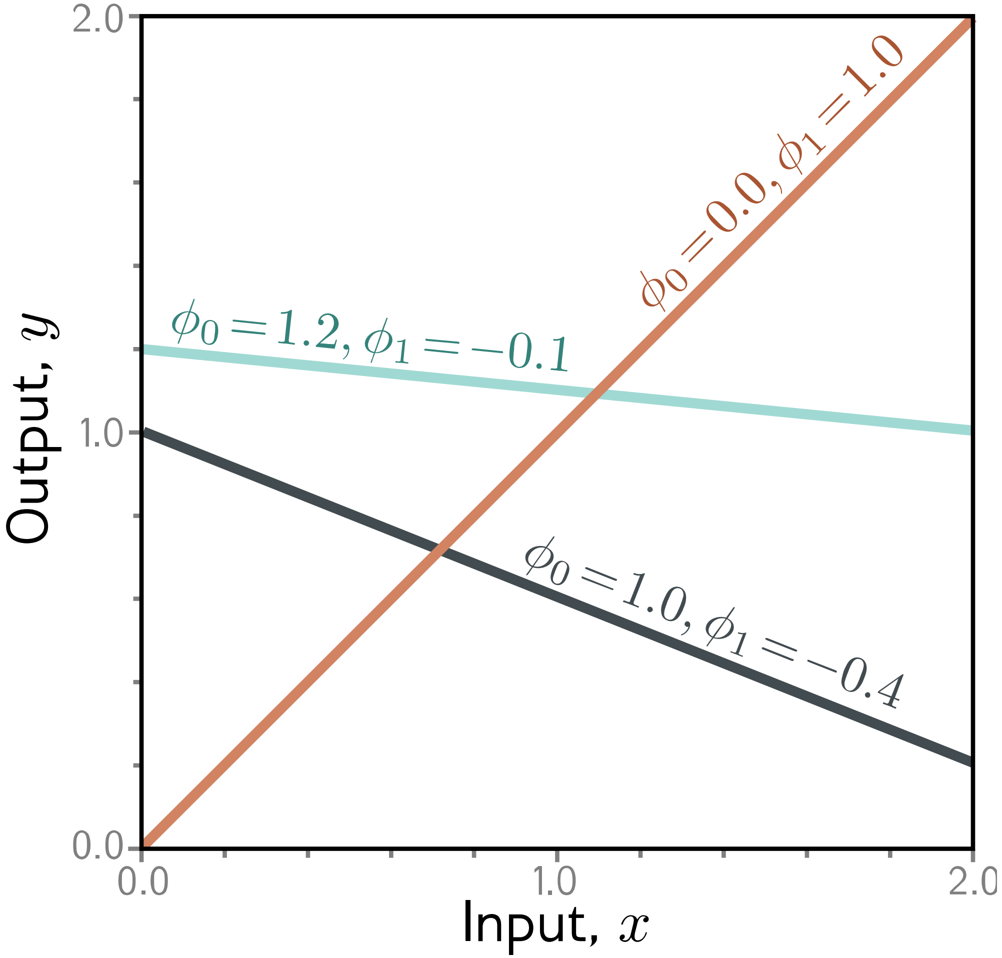
Example: 1D Linear regression model

- Model:

$$
\begin{aligned}
y &= \mathrm{f}[x, \boldsymbol{\phi}] \\
&= \phi_0 + \phi_1 x
\end{aligned}
$$

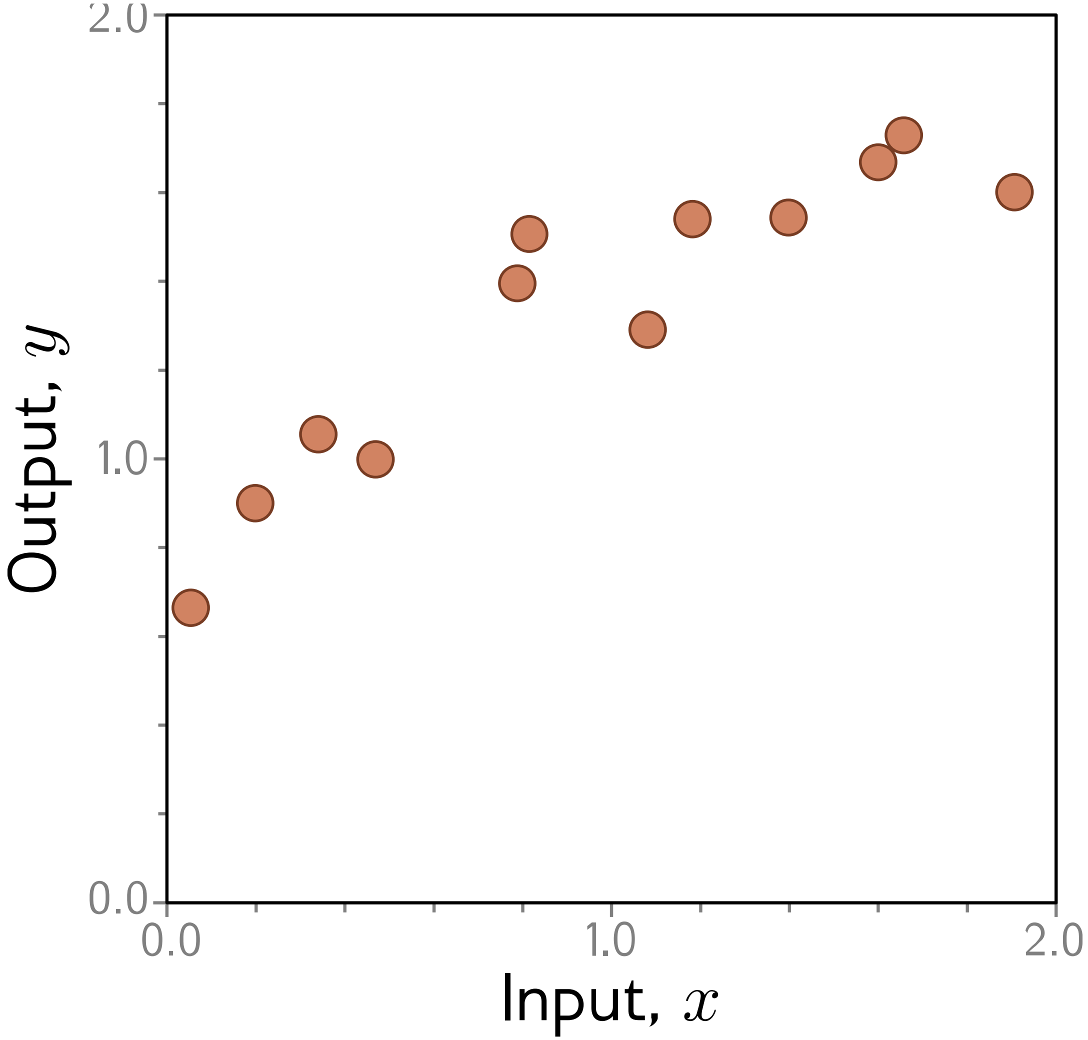
- Parameters

$$
\boldsymbol{\phi} = \begin{bmatrix} \phi_0 \\ \phi_1 \end{bmatrix}
\begin{array}{l} \longleftarrow \text{ y-offset} \\ \longleftarrow \text{ slope} \end{array}
$$

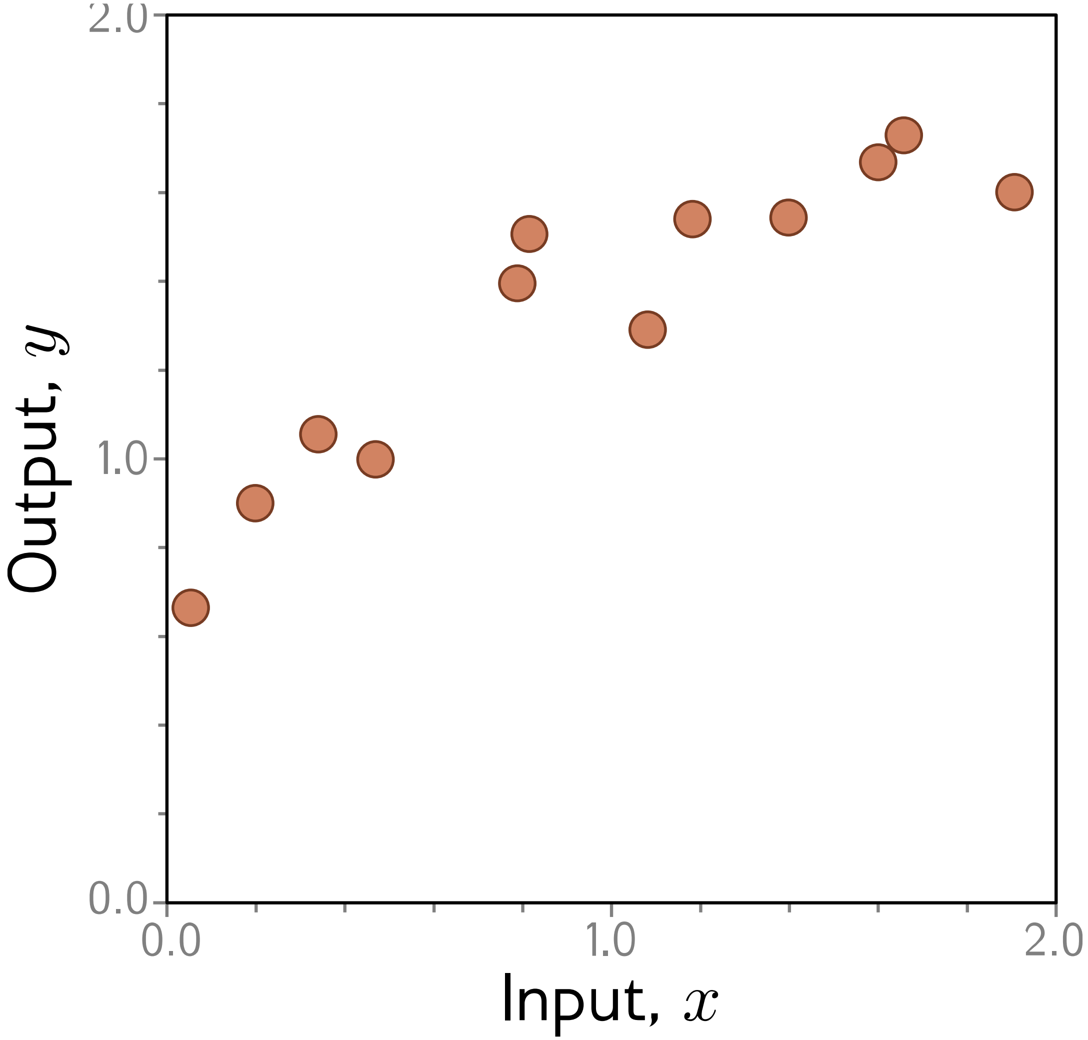


Interactive Figure 2.1

Example: 1D Linear regression training data



Example: 1D Linear regression training data

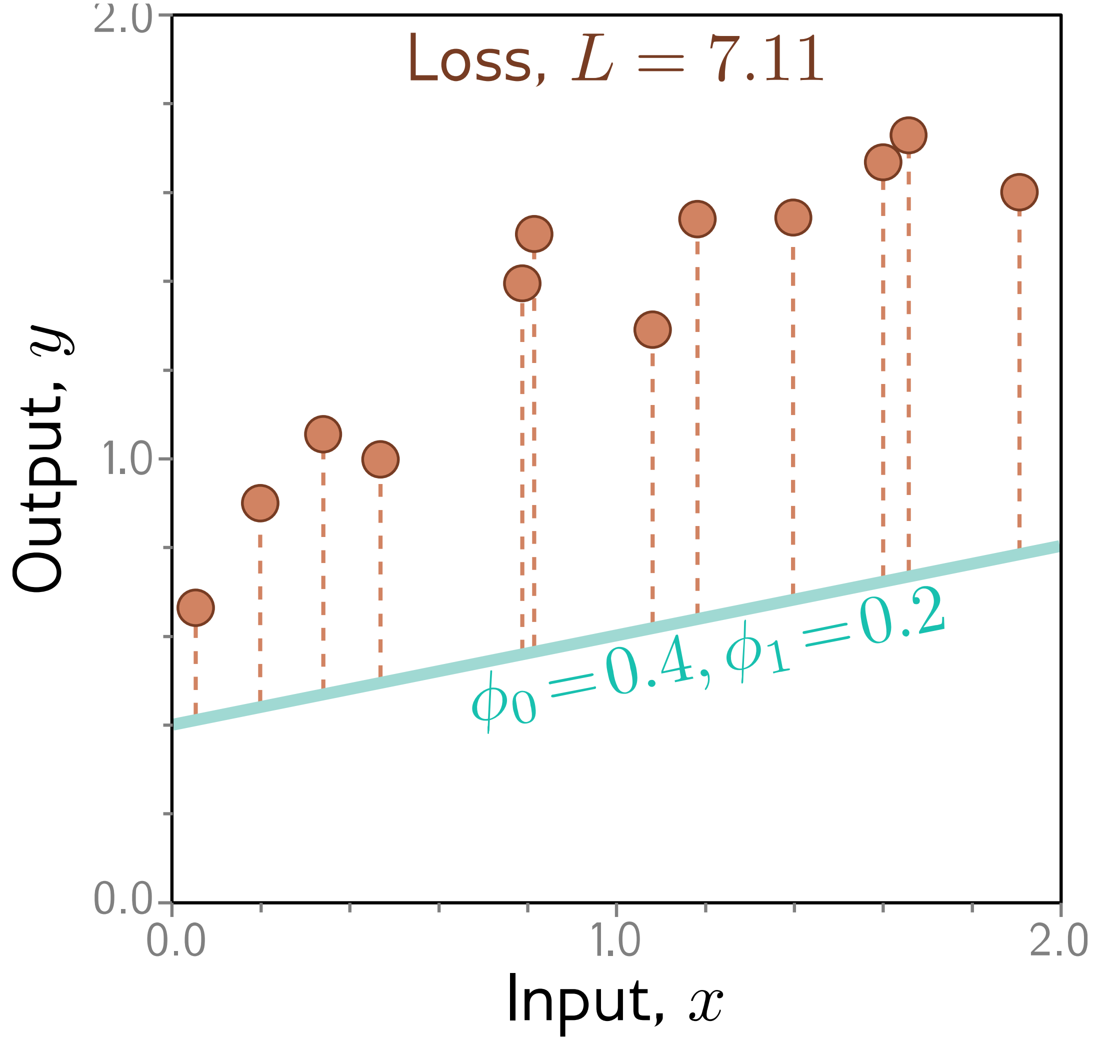


Loss function:

$$
\begin{aligned}
L[\boldsymbol{\phi}] &= \sum_{i=1}^{I} (\mathrm{f}[x_i, \boldsymbol{\phi}] - y_i)^2 \\
&= \sum_{i=1}^{I} (\phi_0 + \phi_1 x_i - y_i)^2
\end{aligned}
$$

"Least squares loss function"

Example: 1D Linear regression loss function



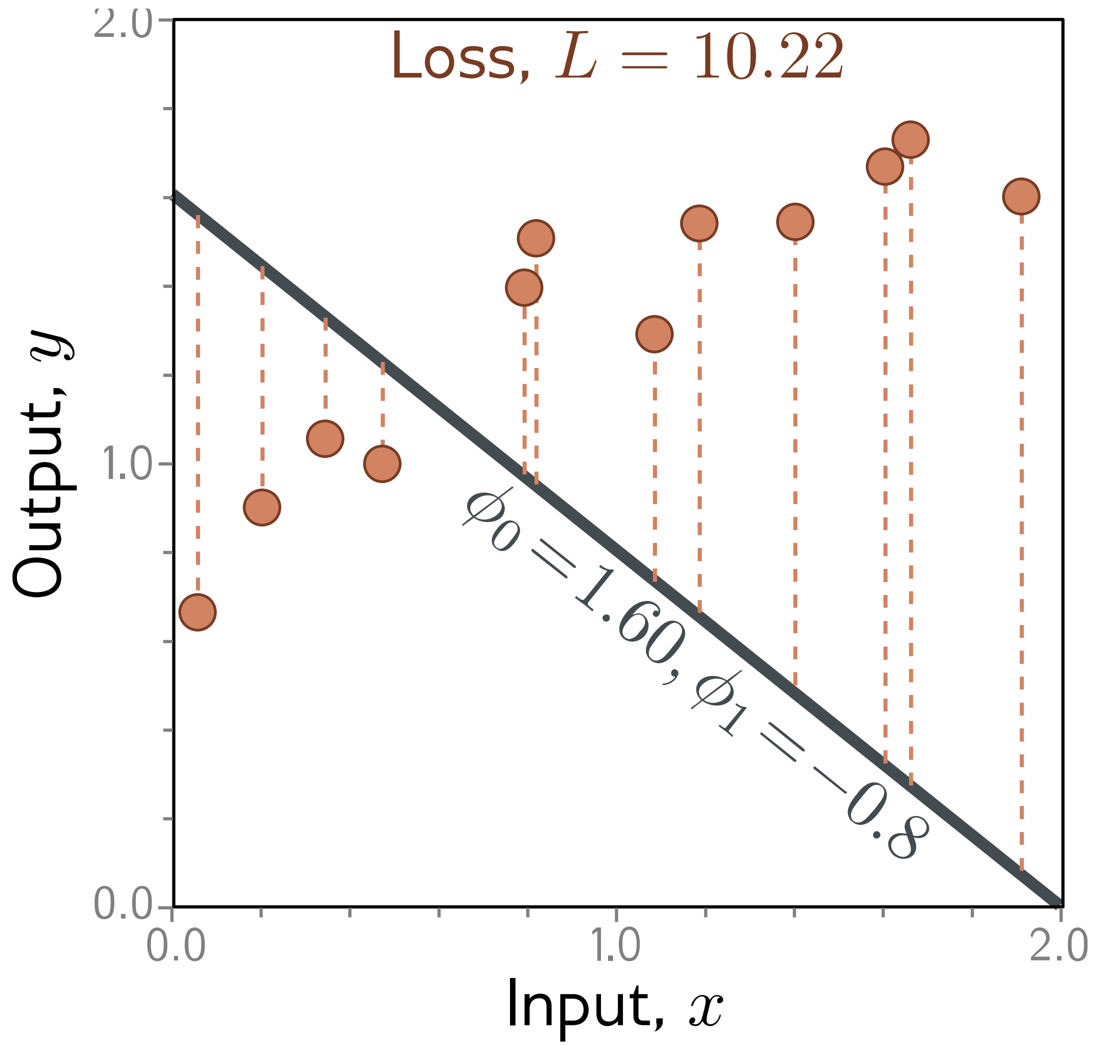
Loss function:

$$L[\phi] = \sum_{i=1}^{I} (\text{f}[x_i, \phi] - y_i)^2$$

$$= \sum_{i=1}^{I} (\phi_0 + \phi_1 x_i - y_i)^2$$

"Least squares loss function"

Example: 1D Linear regression loss function



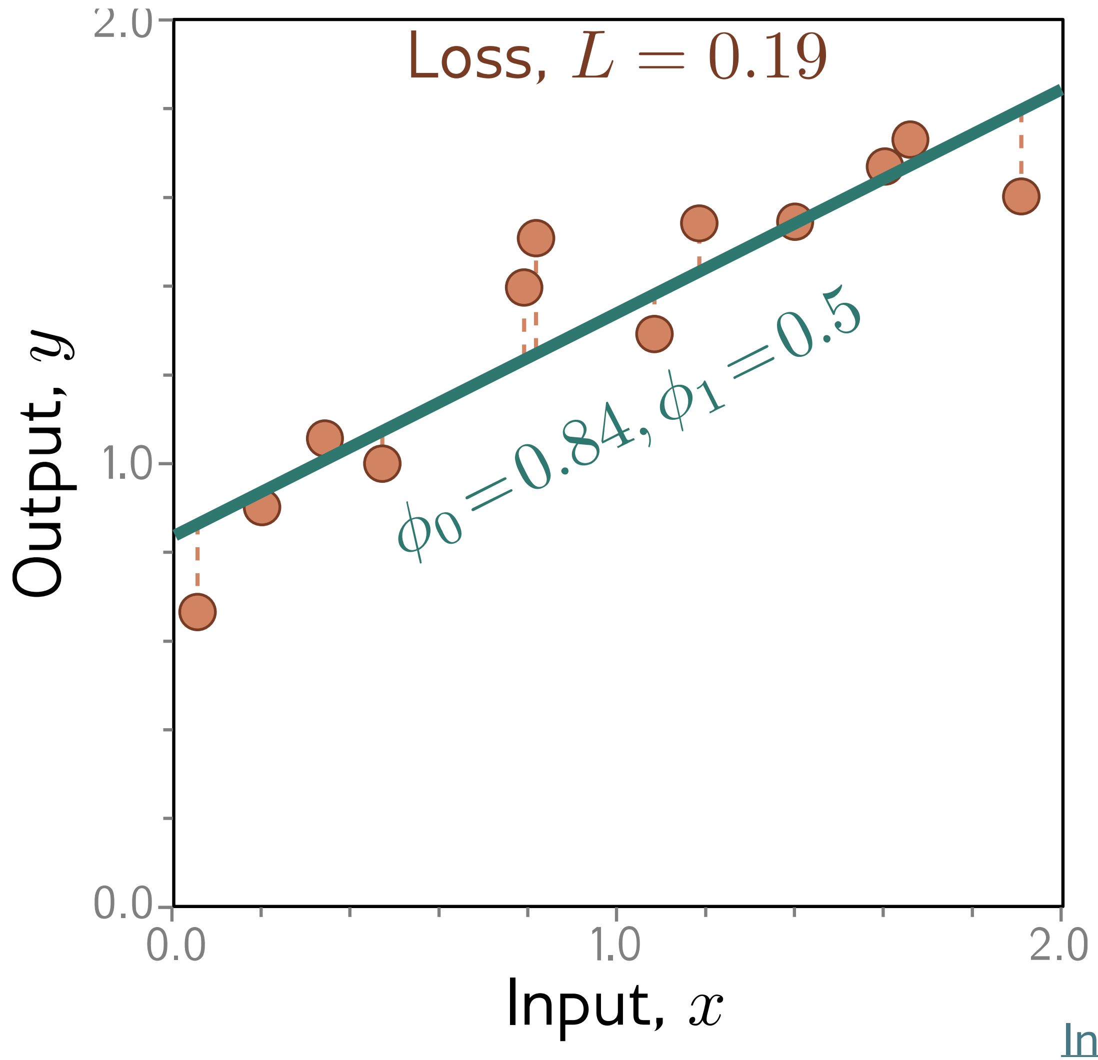
Loss function:

$$L[\phi] = \sum_{i=1}^{I} (\mathrm{f}[x_i, \phi] - y_i)^2$$

$$= \sum_{i=1}^{I} (\phi_0 + \phi_1 x_i - y_i)^2$$

"Least squares loss function"

Example: 1D Linear regression loss function



Loss function:

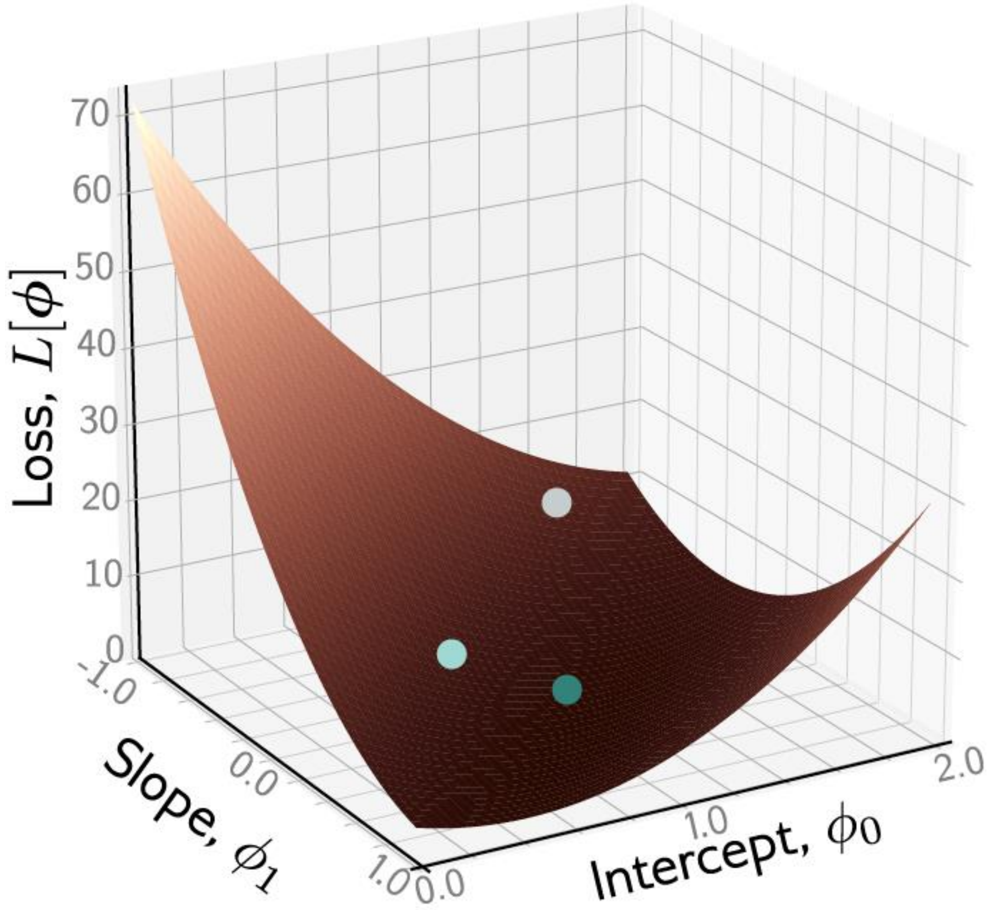
$$L[\boldsymbol{\phi}] = \sum_{i=1}^{I} (\mathrm{f}[x_i, \boldsymbol{\phi}] - y_i)^2$$

$$= \sum_{i=1}^{I} (\phi_0 + \phi_1 x_i - y_i)^2$$

"Least squares loss function"

Interactive Figure 2.2

Example: 1D Linear regression loss function

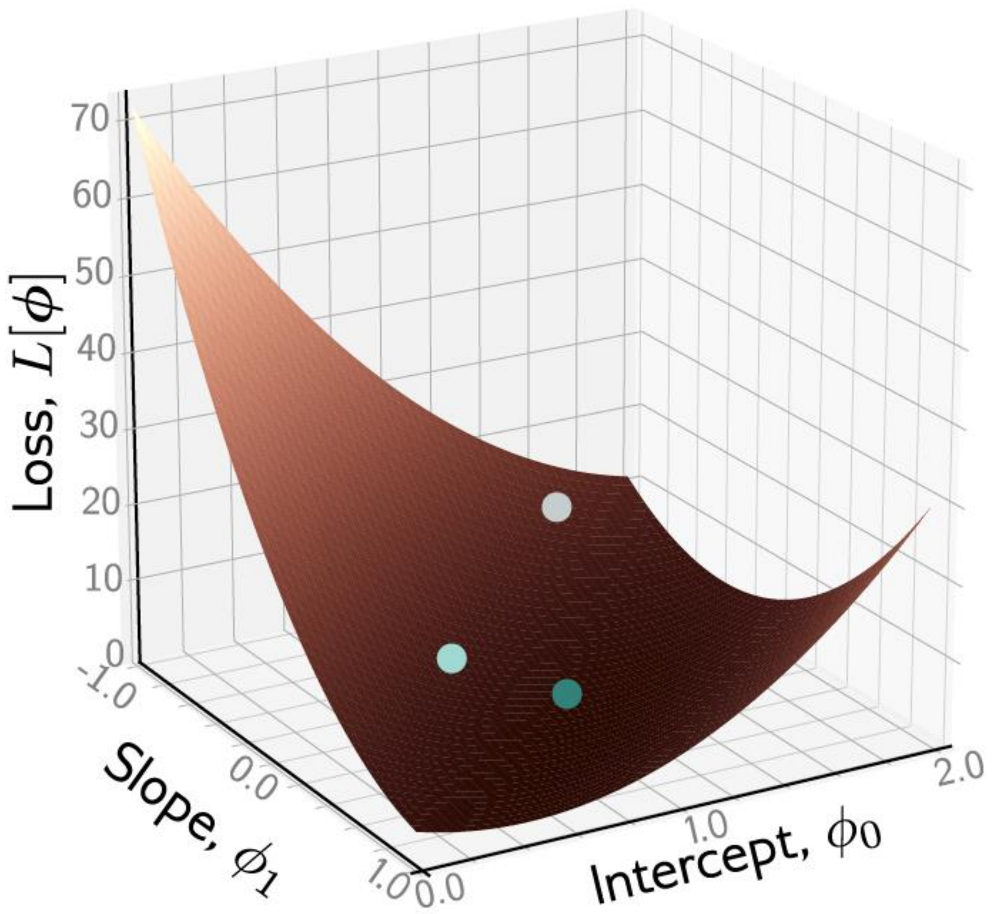


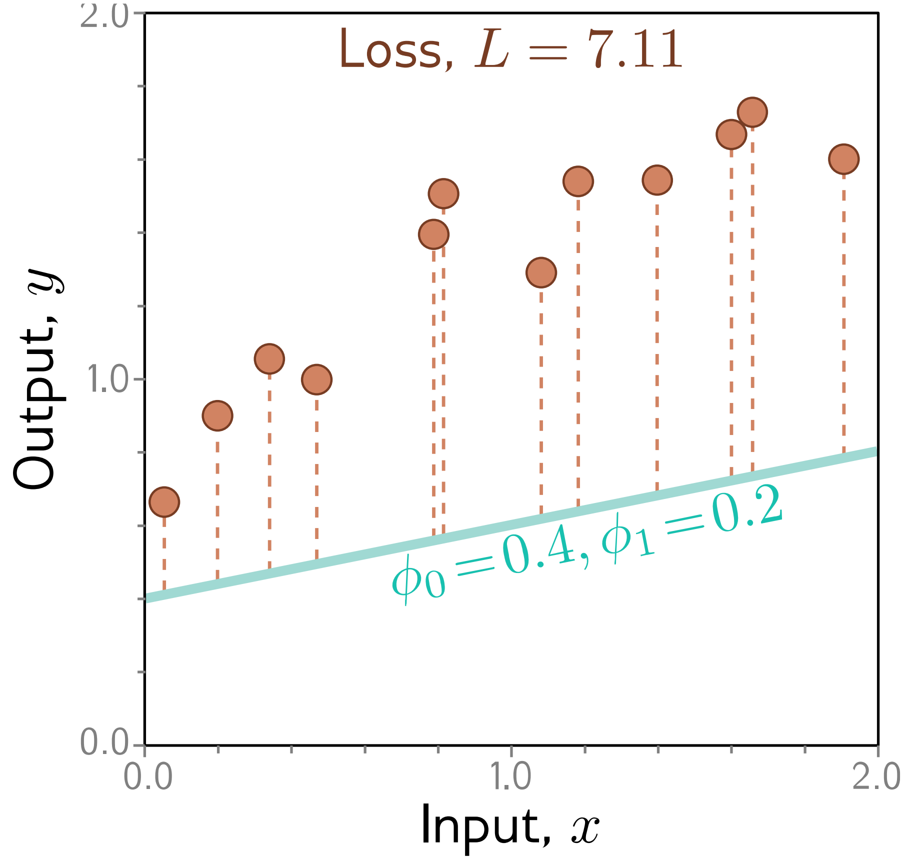
Loss function:

$$L[\phi] = \sum_{i=1}^{I} (\mathrm{f}[x_i, \phi] - y_i)^2$$
$$= \sum_{i=1}^{I} (\phi_0 + \phi_1 x_i - y_i)^2$$

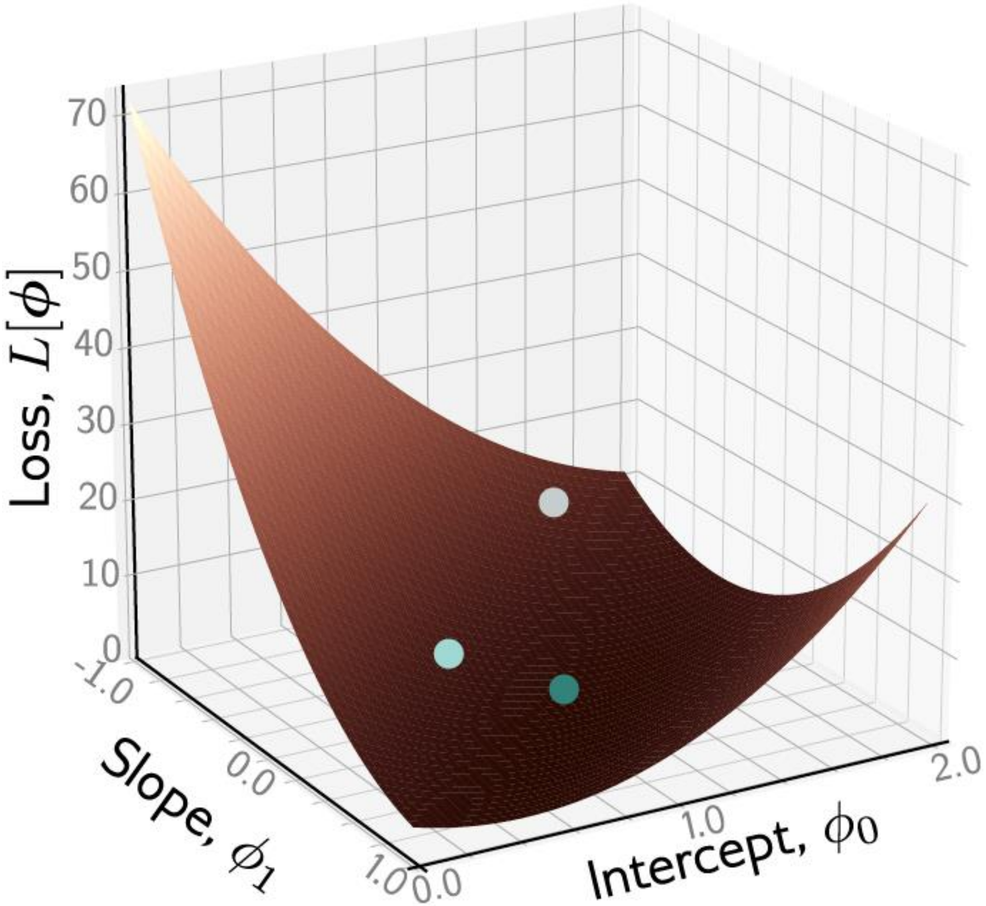
"Least squares loss function"

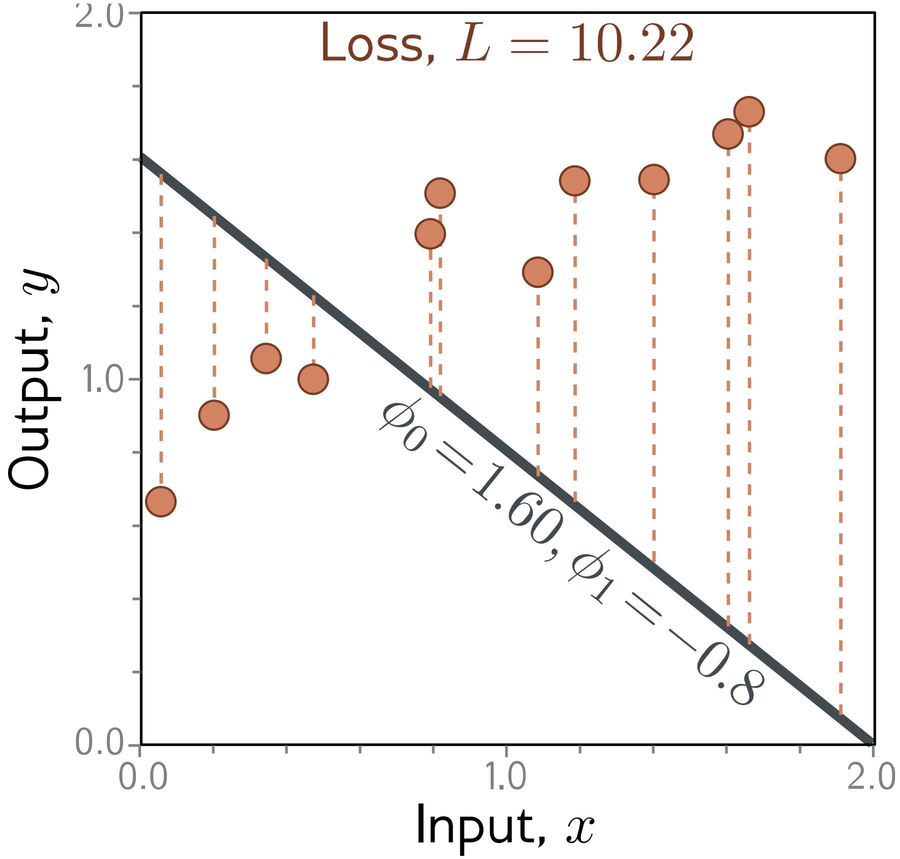
Example: 1D Linear regression loss function



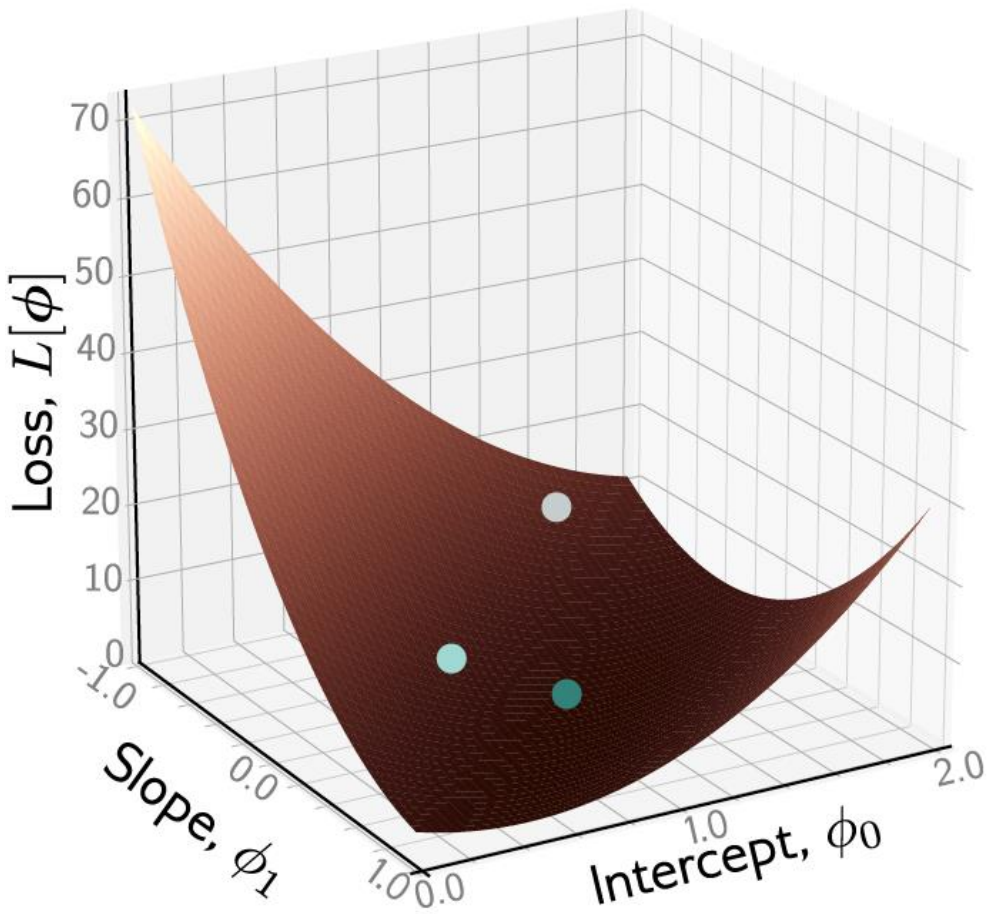


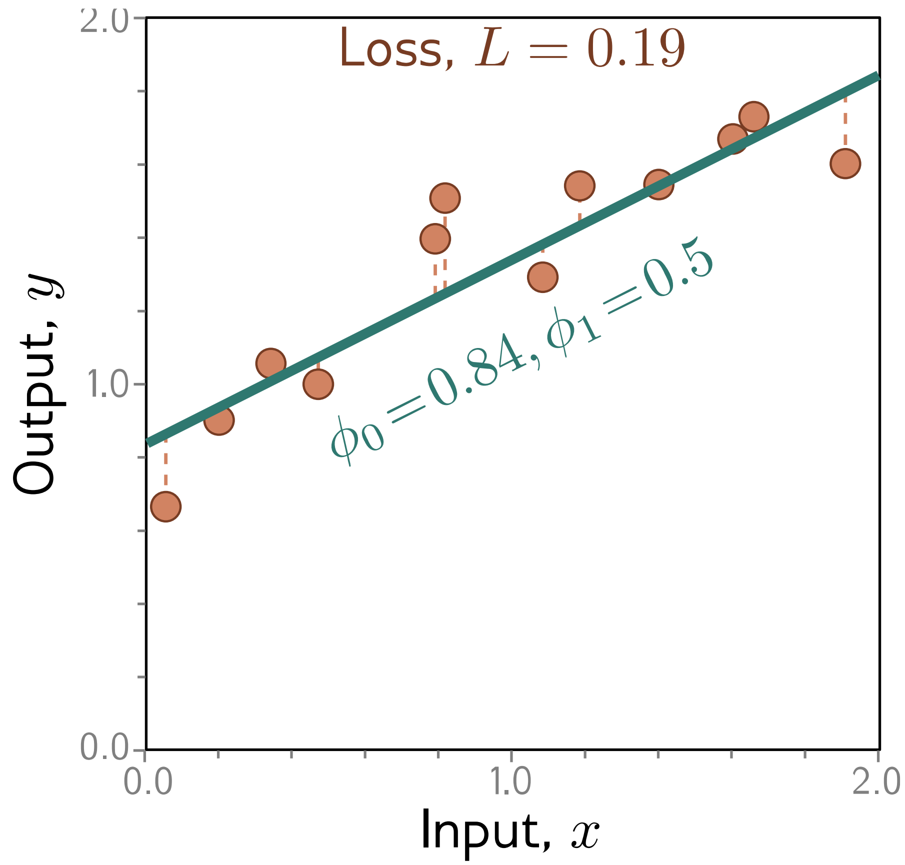
Example: 1D Linear regression loss function



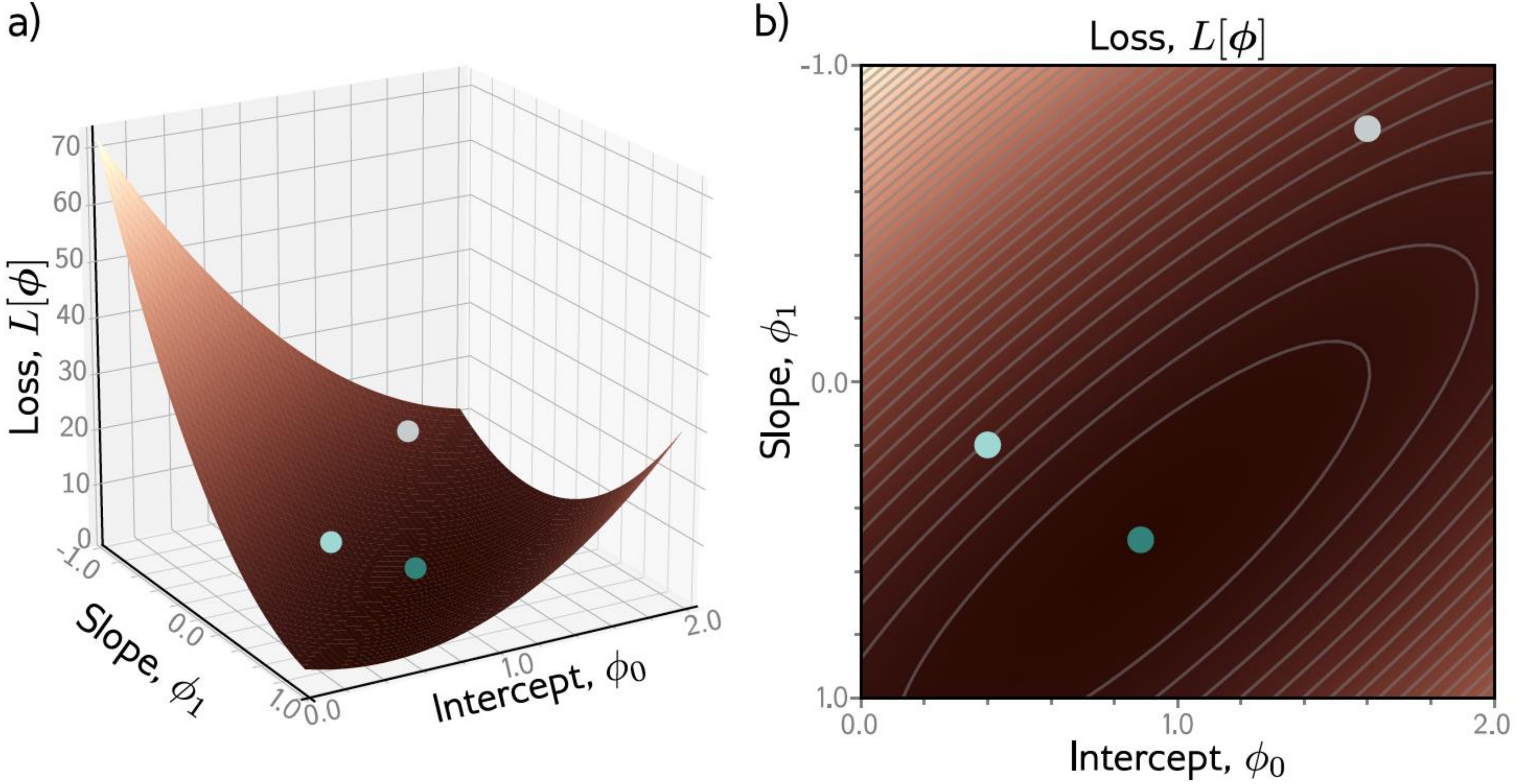


Example: 1D Linear regression loss function

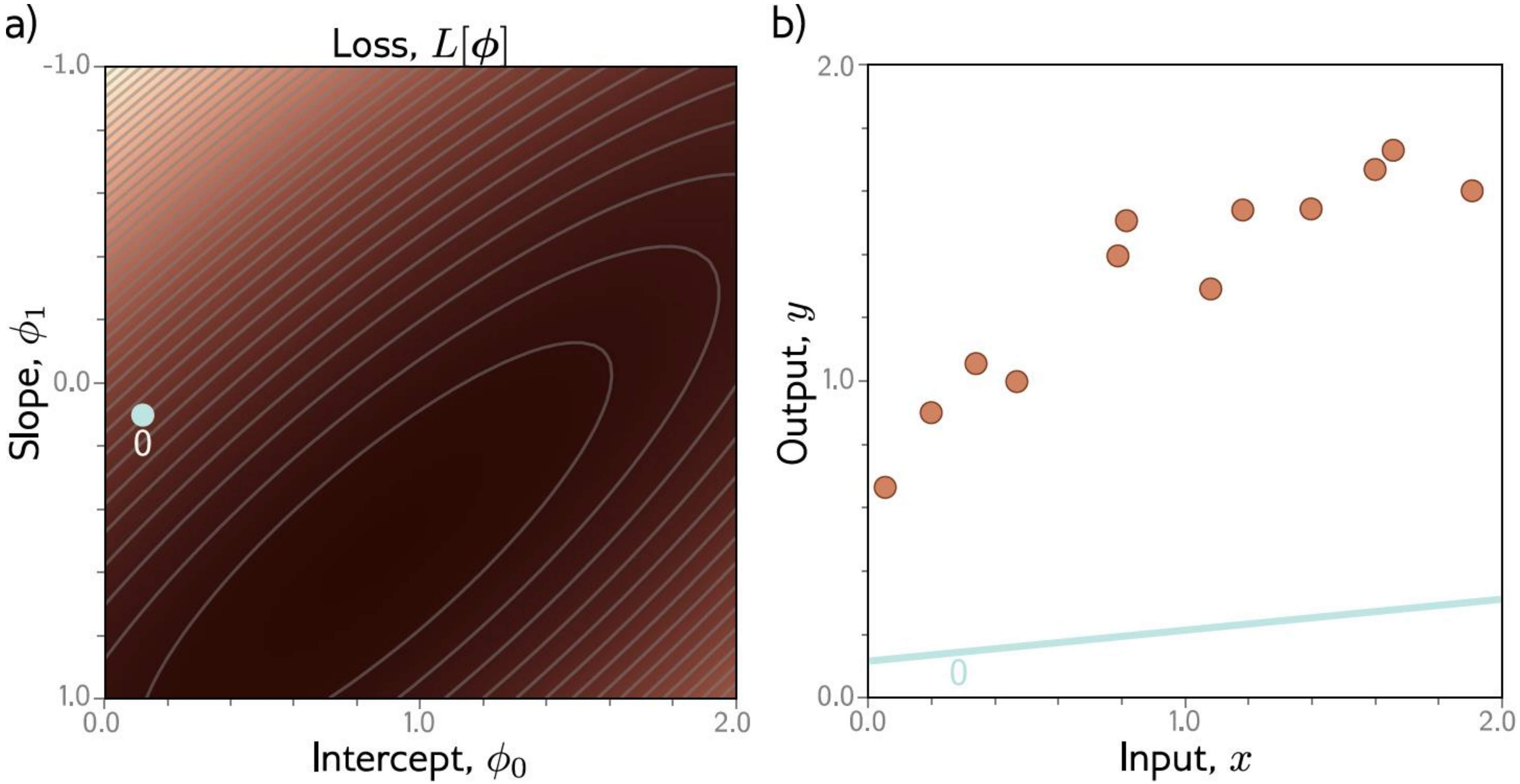




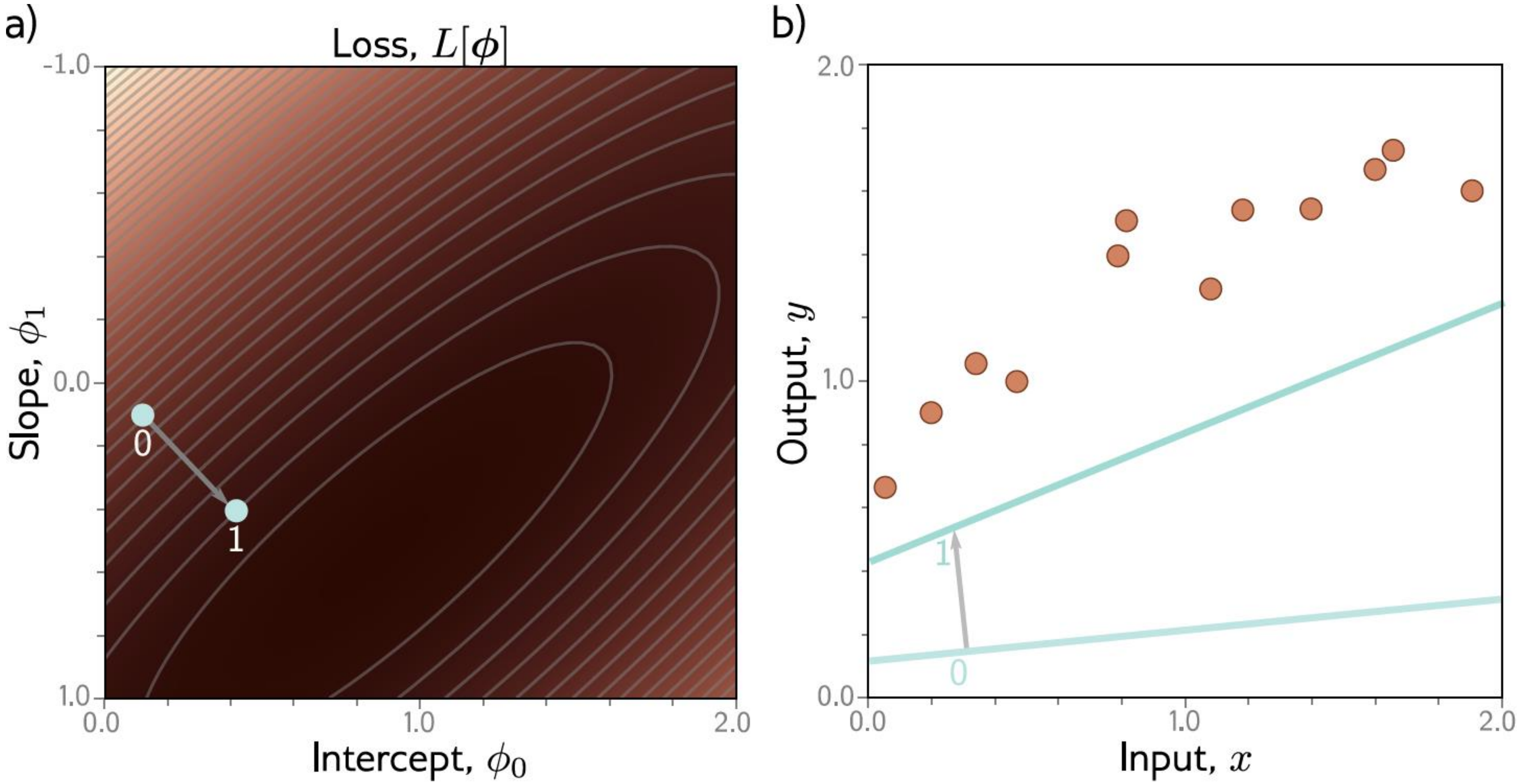
Example: 1D Linear regression loss function



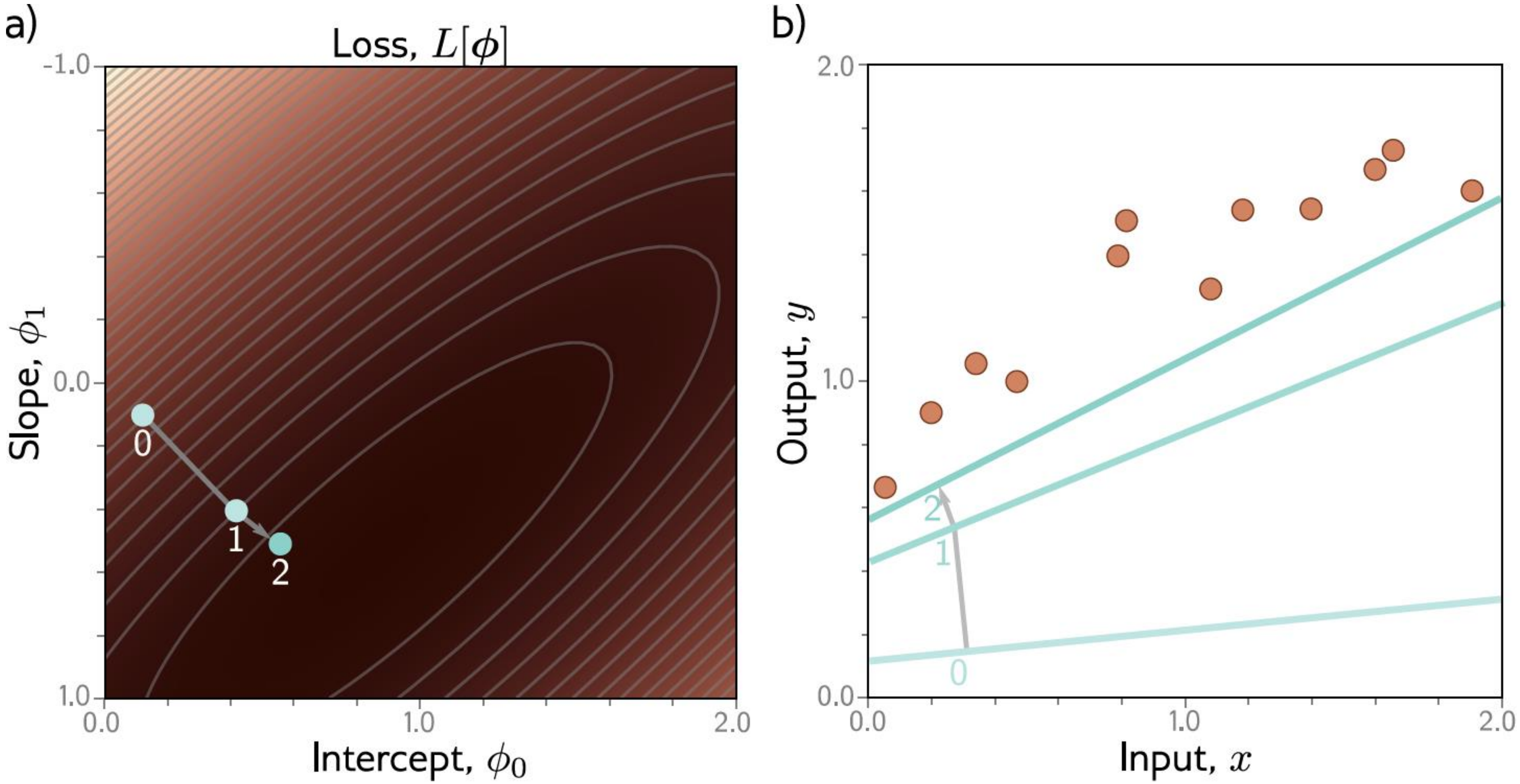
Example: 1D Linear regression training



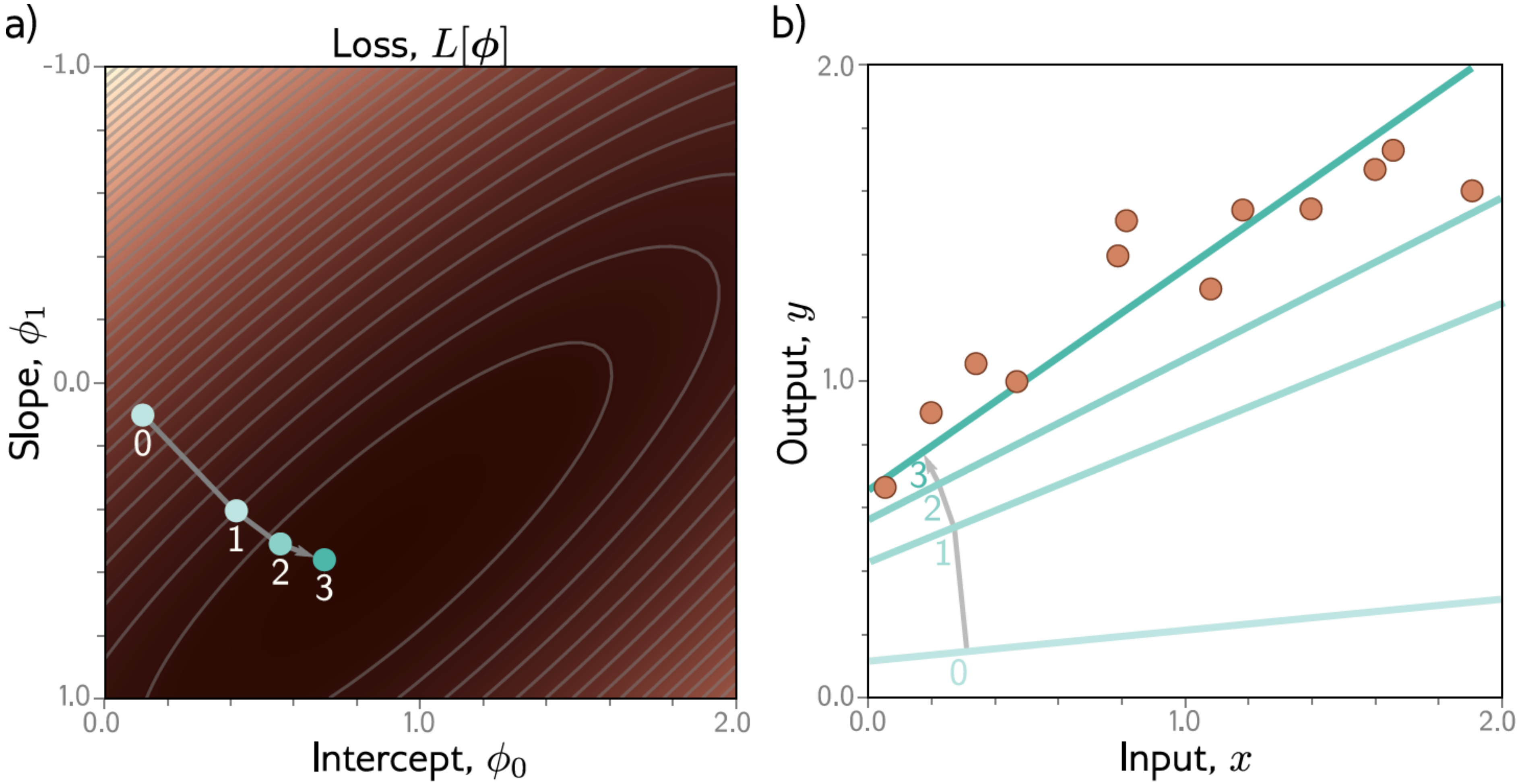
Example: 1D Linear regression training



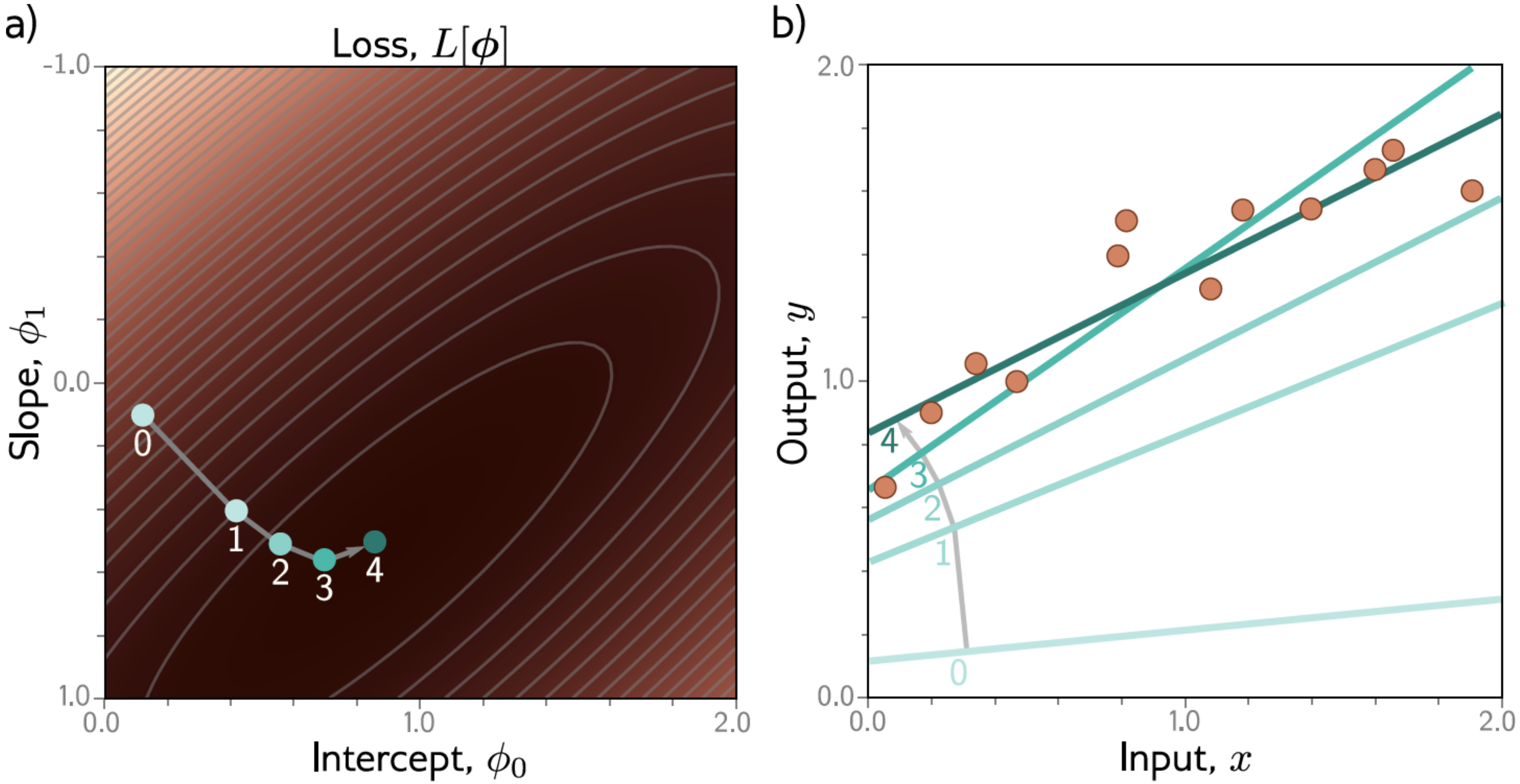
Example: 1D Linear regression training



Example: 1D Linear regression training



Example: 1D Linear regression training

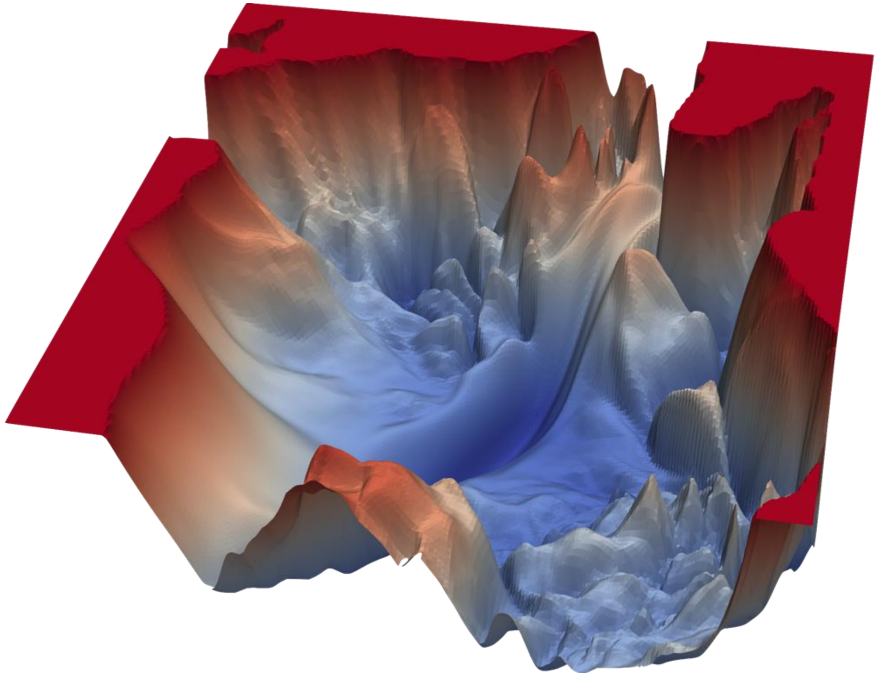


This technique is known as gradient descent

[Interactive Figure 2.3]

Possible objections

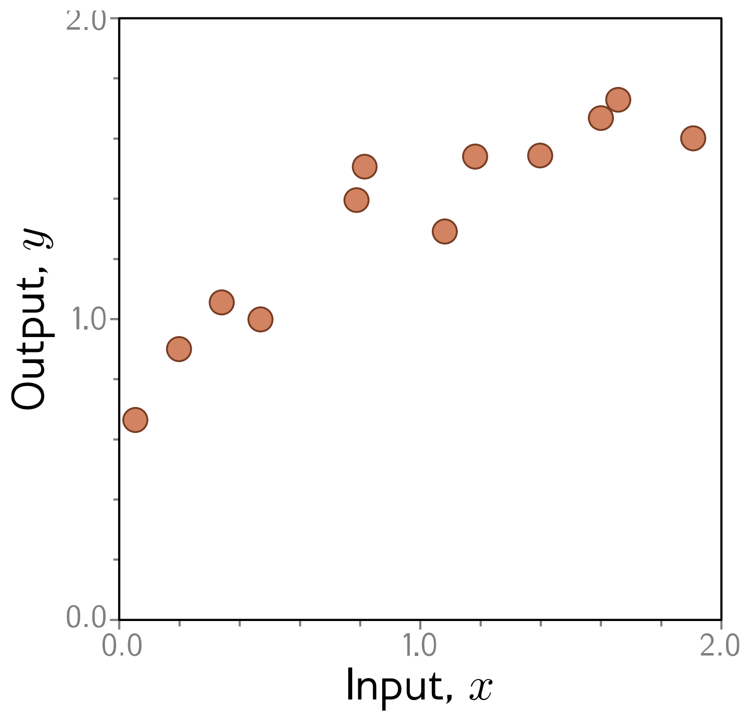
- But you can fit the line model in closed form!
  - Yes – but we won’t be able to do this for more complex models
- But we could exhaustively try every slope and intercept combo!
  - Yes – but we won’t be able to do this when there are a million parameters



Here's a visualization of the loss surface for the 56-layer neural network VGG-56(from Visualizing the Loss Landscape of Neural Networks -- https://losslandscape.com/explorer

Example: 1D Linear regression testing

- Test with different set of paired input/output data (Test Set)
  - Measure performance
  - Degree to which *Loss* is same as training = generalization
- Might not generalize well because
  - Model too simple: underfitting
  - Model too complex
    - fits to statistical peculiarities of data
    - this is known as overfitting



Any Questions?

Next Lecture

- How do we choose a loss function in a principled way?