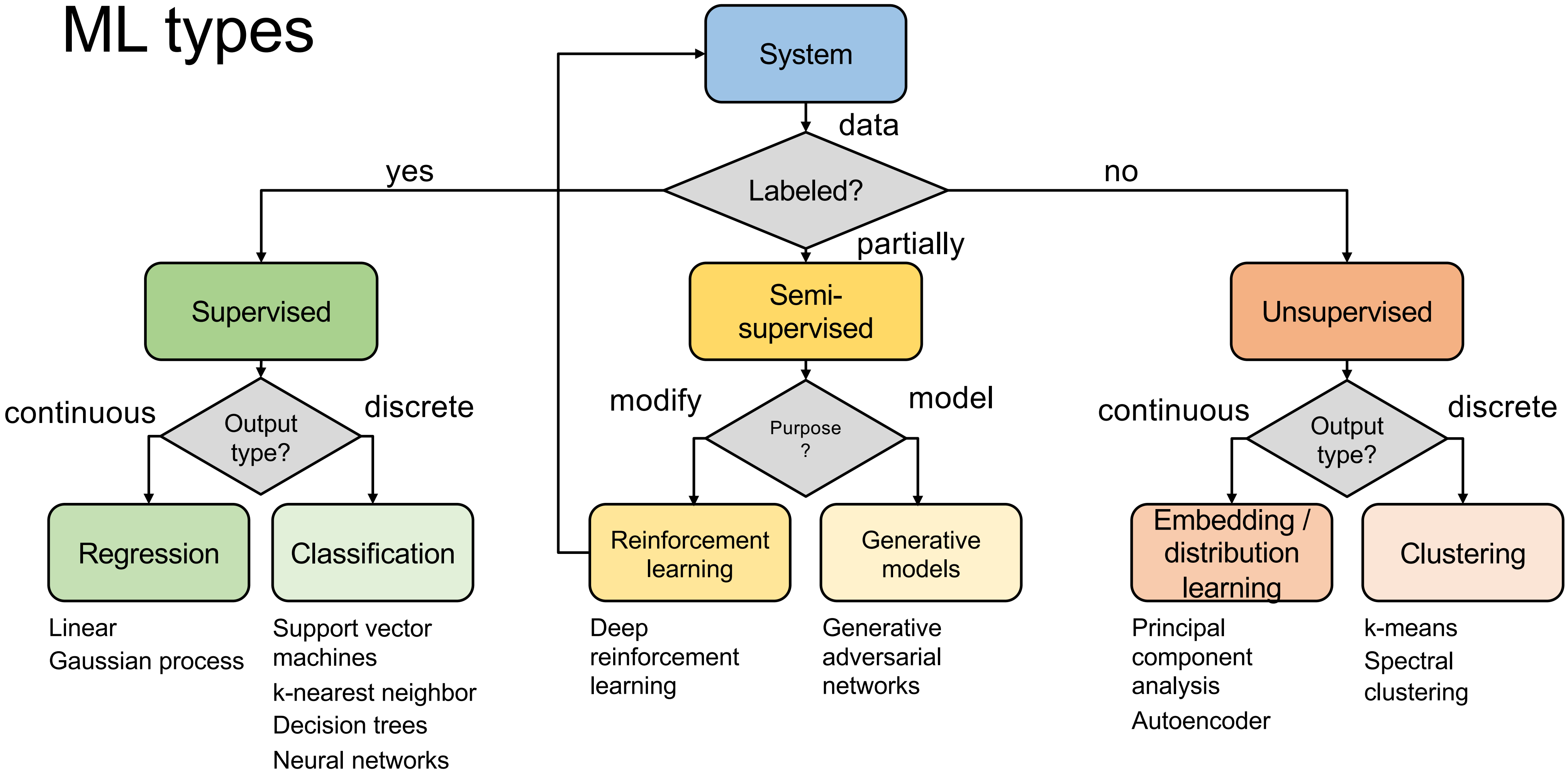
Machine Learning in Biomedical Sciences and Bioengineering

Lecture 2
Regression

2025 version 1.00

James Choi

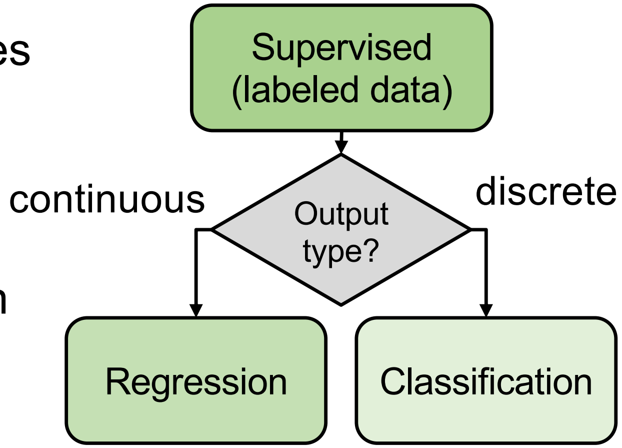
ML types


System
data
Labeled?
yes
no
partially
Supervised
Semi-supervised
Unsupervised
continuous
Output type?
discrete
modify
Purpose ?
model
continuous
Output type?
discrete
Regression
Classification
Reinforcement learning
Generative models
Embedding / distribution learning
Clustering
Linear
Gaussian process
Support vector machines
k-nearest neighbor
Decision trees
Neural networks
Deep reinforcement learning
Generative adversarial networks
Principal component analysis
Autoencoder
k-means
Spectral clustering

Supervised learning

- Regression
  - Predicts a range of values
  - **Continuous** range of possible outputs

  - Predict the concentration of drugs in the liver for a given systemic dose.



- Classification
  - Predicts categories
  - **Discrete** number of possible outputs

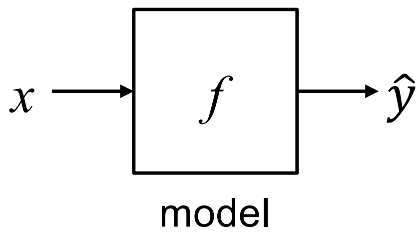
  - Predict whether the imaged mass is a cyst or malignant tumour

General machine learning process

1. **Determine the objective**
   What do we want to achieve?
2. **Prepare the data**
   What kind of data do we need to inform the model?
3. **Design an architecture**
   How should we construct the ML model?
4. **Create a loss function**
   How do we inform the model that its performance is good or bad?
5. **Optimise the architecture**
   How do we adjust the model parameters to become better?

Determine the objective

- What do we want to achieve?



- **Example:** Predict the concentration of drugs that enters the brain from the amount of drugs that are administered in the bloodstream.

Prepare the data

x: input variable (feature)

y: output variable (targets, true values)

i: sample number

m: total number of training samples

$(x, y) =$ a training sample

$(x^{(i)}, y^{(i)}) =$ the i^{th} training example

i	x	y
Sample Number	**Mass of drugs injected (mg)**	**Mass of drugs in brain (µg)**
1	0.5	174.7
2	1.0	375.3
3	1.5	611.1
4	2.0	243.9
5	2.5	956.7
6	3.0	844.3
7	3.5	1165.6
8	4.0	885.9
9	4.5	1059.5
10	5.0	865.7
11	5.5	1057.8
12	6.0	1522.1
13	6.5	1373.1
14	7.0	1898.9
15	7.5	1413.6
16	8.0	2166.3
17	8.5	1930
18	9.0	2155.2
19	9.5	2474.6
20	10.0	2040.9

Prepare the data

x: input variable (feature)

y: output variable (targets, true values)

i: sample number

m: total number of training samples

(x, y) = a training sample

$(x^{(i)}, y^{(i)})$ = the i^{th} training example

$(x^{(3)}, y^{(3)}) = (1.5, 611.1)$

i	x	y
Sample Number	Mass of drugs injected (mg)	Mass of drugs in brain (μg)
1	0.5	174.7
2	1.0	375.3
3	1.5	611.1
4	2.0	243.9
5	2.5	956.7
6	3.0	844.3
7	3.5	1165.6
8	4.0	885.9
9	4.5	1059.5
10	5.0	865.7
11	5.5	1057.8
12	6.0	1522.1
13	6.5	1373.1
14	7.0	1898.9
15	7.5	1413.6
16	8.0	2166.3
17	8.5	1930
18	9.0	2155.2
19	9.5	2474.6
20	10.0	2040.9

Prepare the data

x: input variable (feature)

y: output variable (targets, true values)

i: sample number

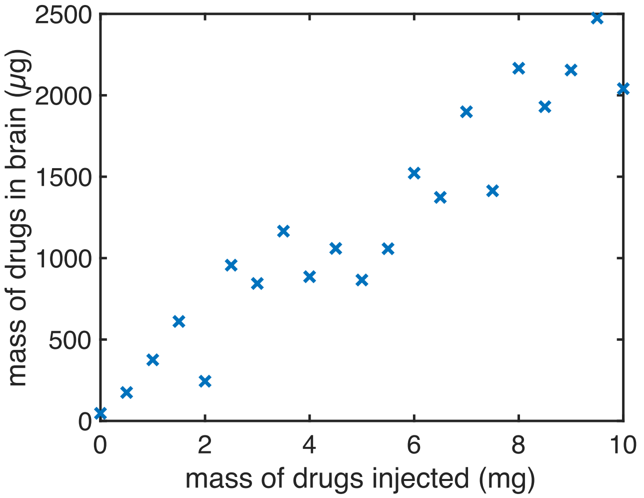
m: total number of training samples

(x, y) = a training sample

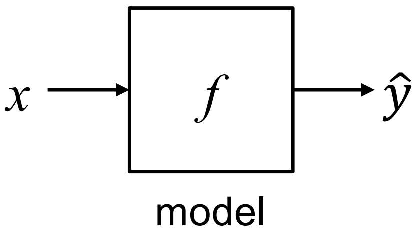
$(x^{(i)}, y^{(i)})$ = the i^{th} training example

$(x^{(3)}, y^{(3)}) = (1.5, 611.1)$

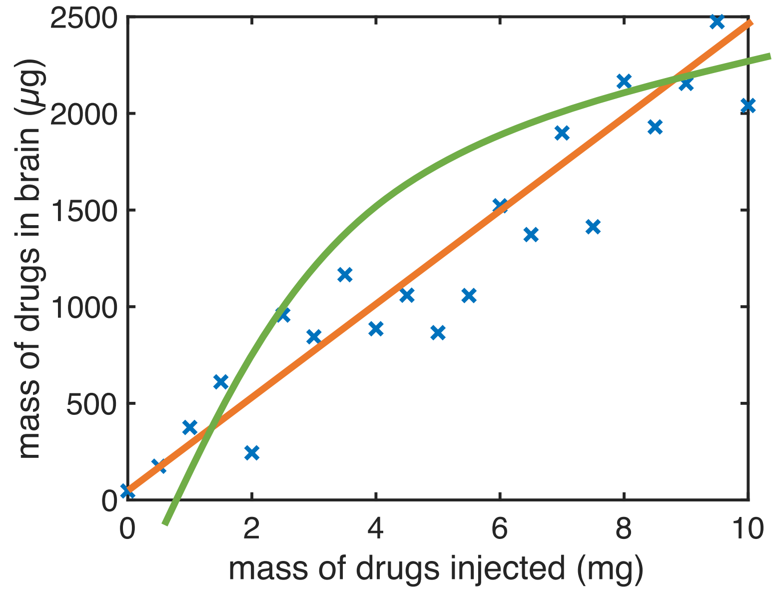
i	x	y
Sample Number	Mass of drugs injected (mg)	Mass of drugs in brain (µg)
1	0.5	174.7
2	1.0	375.3
3	1.5	611.1
4	2.0	243.9
5	2.5	956.7
6	3.0	844.3
7	3.5	1165.6
8	4.0	885.9
9	4.5	1059.5
10	5.0	865.7
11	5.5	1057.8
12	6.0	1522.1
13	6.5	1373.1
14	7.0	1898.9
15	7.5	1413.6
16	8.0	2166.3
17	8.5	1930
18	9.0	2155.2
19	9.5	2474.6
20	10.0	2040.9



Design the architecture



- What **model architecture** would work best for predicting the output based on the input available?



Consider the following candidate models:

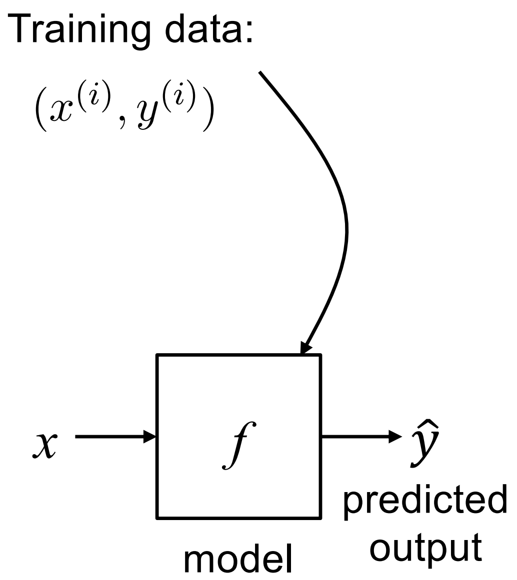
$$\boxed{y = wx + b}$$

$$y = w_1 x + w_2 x^2 + b$$

$$y = w_1 x + w_2 x^2 + w_3 x^2 + b$$

$$y = w log_{10}(x) + b$$

Design the architecture



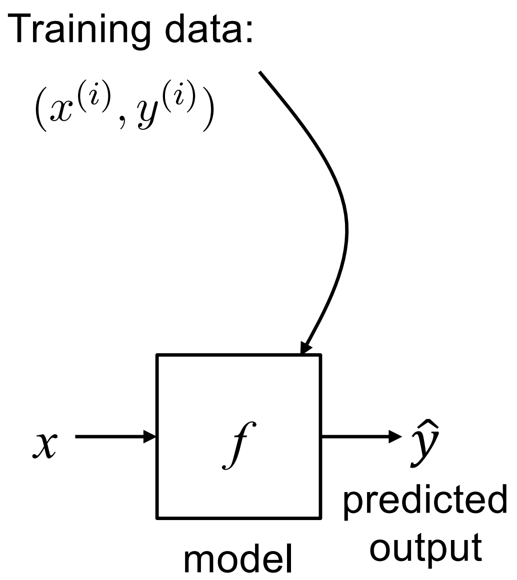
$$y = wx + b$$

w is the weight
b is the bias

$$f_{w,b}(x) = wx + b$$

$$f(x) = wx + b$$

Create a loss function



$$\hat{y}^{(i)} = f_{w,b}(x^{(i)})$$

$$f_{w,b}(x^{(i)}) = wx^{(i)} + b$$

$$y = wx + b$$

$$y = w_1 x + w_2 x^2 + w_3 x^2 + b$$

$$y = w_1 x_1 + w_2 x_2 + w_3 x_3 + b$$

$$y = w log_{10}(x) + b$$

Find w, b:

$\hat{y}^{(i)}$ is close to $y^{(i)}$ for all $(x^{(i)}, y^{(i)})$

How can we measure how well our guesses for w and b are?

$$\text{error} = \hat{y}^{(i)} - y^{(i)}$$

$$\text{or...} = (\hat{y}^{(i)} - y^{(i)})^2$$

Create a loss function

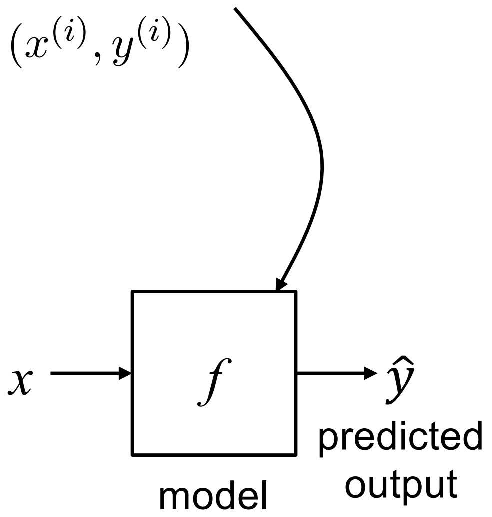
$$f_{w,b}(x) = wx + b$$

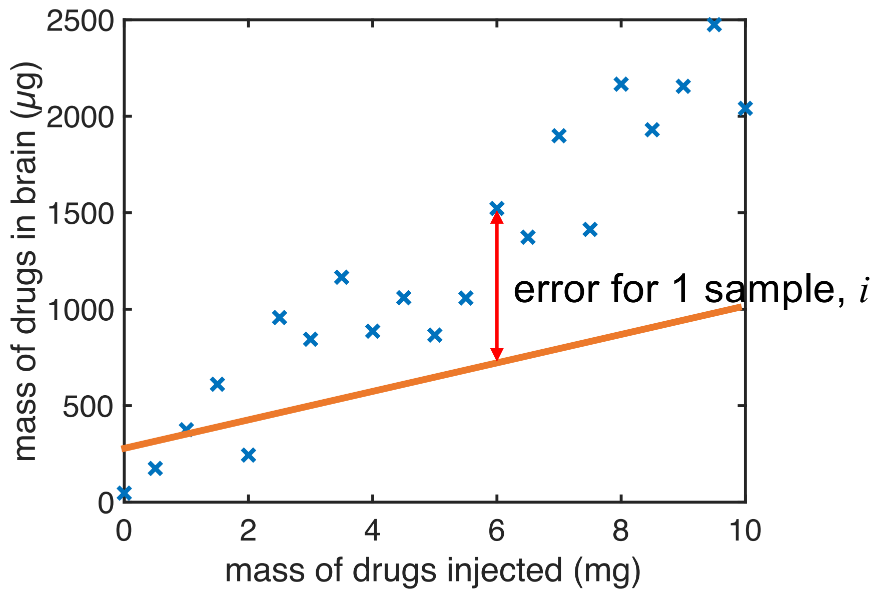
$$\text{error} = \hat{y}^{(i)} - y^{(i)}$$

$$\text{or...} = (\hat{y}^{(i)} - y^{(i)})^2$$

Training data:

$(x^{(i)}, y^{(i)})$



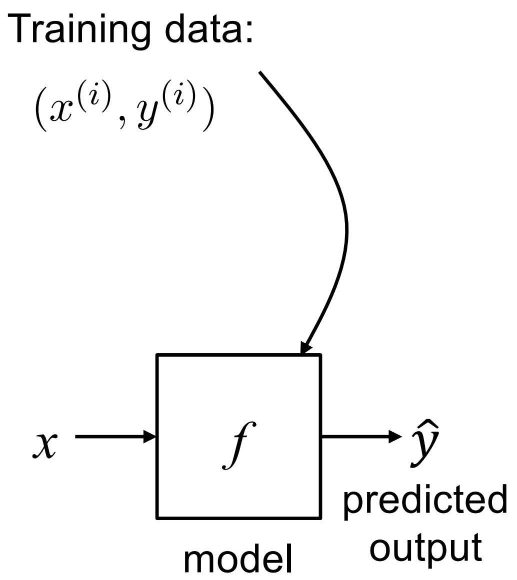


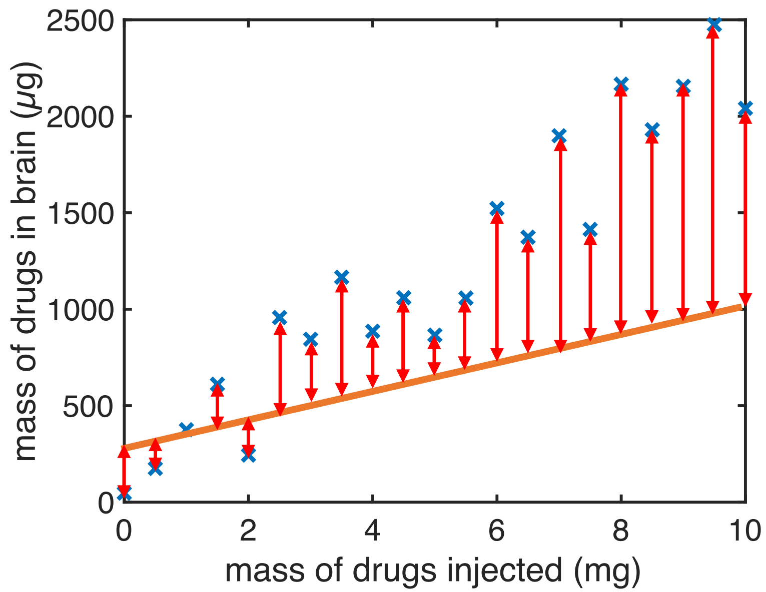
Create a loss function

$$f_{w,b}(x) = wx + b$$

$$\text{error} = \hat{y}^{(i)} - y^{(i)}$$

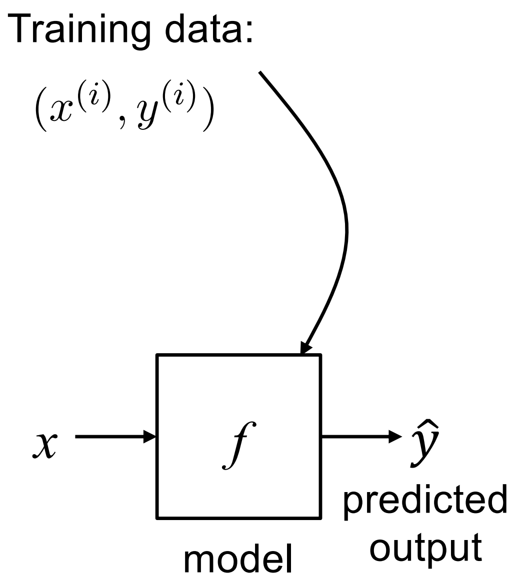
$$\text{or...} = (\hat{y}^{(i)} - y^{(i)})^2$$





Need to account for all sample errors

Create a loss function



Squared error cost function:

$$J(w,b) = \frac{1}{2m} \sum_{i=1}^{m} \left(\hat{y}^{(i)} - y^{(i)}\right)^2$$

$$J(w,b) = \frac{1}{2m} \sum_{i=1}^{m} \left(f_{w,b}(x^{(i)}) - y^{(i)}\right)^2$$

$$J(w,b) = \frac{1}{2m} \sum_{i=1}^{m} \left(wx^{(i)} + b - y^{(i)}\right)^2$$

$$\underset{w,b}{\text{minimize}}\ J(w,b)$$

adjust w and b so that the cost function is minimal.

$$\boxed{\underset{w_1,w_2,\ldots w_n,b}{\text{minimize}}\ J(w_1, w_2, \ldots w_n, b)}$$

Overview

Model: $f_{w,b}(x) = wx + b$

Parameters: w, b

Cost function: $$J(w,b) = \frac{1}{2m}\sum_{i=1}^{m}\left(f_{w,b}(x^{(i)}) - y^{(i)}\right)^2$$

Optimisation goal: $\underset{w,b}{\text{minimize}}\, J(w,b)$

Optimise the architecture

$$\underset{w,b}{\text{minimize}}\, J(w,b)$$

$$w_{i+1} = w_i - \text{ something}$$

$$b_{i+1} = b_i - \text{ something}$$

$$w_{i+1} = w_i - \alpha \frac{\partial}{\partial w} J(w_i, b_i)$$

$$b_{i+1} = b_i - \alpha \frac{\partial}{\partial b} J(w_i, b_i)$$

$$\begin{aligned}
\frac{\partial}{\partial w} J(w,b) &= \frac{\partial}{\partial w}\left(\frac{1}{2m}\sum_{i=1}^{m}\left(f_{w,b}(x^{(i)}) - y^{(i)}\right)^2\right) \\
&= \frac{\partial}{\partial w}\left(\frac{1}{2m}\sum_{i=1}^{m}\left(wx^{(i)} + b - y^{(i)}\right)^2\right) \\
&= \frac{1}{2m}\sum_{i=1}^{m}\left(wx^{(i)} + b - y^{(i)}\right) 2x^{(i)} \\
&= \frac{1}{m}\sum_{i=1}^{m}\left(wx^{(i)} + b - y^{(i)}\right) x^{(i)}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial}{\partial b} J(w,b) &= \frac{\partial}{\partial b}\left(\frac{1}{2m}\sum_{i=1}^{m}\left(f_{w,b}(x^{(i)}) - y^{(i)}\right)^2\right) \\
&= \frac{\partial}{\partial b}\left(\frac{1}{2m}\sum_{i=1}^{m}\left(wx^{(i)} + b - y^{(i)}\right)^2\right) \\
&= \frac{1}{2m}\sum_{i=1}^{m}\left(wx^{(i)} + b - y^{(i)}\right) 2 \\
&= \frac{1}{m}\sum_{i=1}^{m}\left(wx^{(i)} + b - y^{(i)}\right)
\end{aligned}$$

Optimise the architecture

Gradient descent

$$\underset{w,b}{\text{minimize}}\; J(w,b)$$

$$w_{i+1} = w_i - \alpha \frac{\partial}{\partial w} J(w_i, b_i) \qquad\qquad b_{i+1} = b_i - \alpha \frac{\partial}{\partial b} J(w_i, b_i)$$

$$\frac{\partial}{\partial w} J(w,b) = \frac{1}{m}\sum_{i=1}^{m}\left(f_{w,b}(x^{(i)}) - y^{(i)}\right) x^{(i)} \qquad\qquad \frac{\partial}{\partial b} J(w,b) = \frac{1}{m}\sum_{i=1}^{m}\left(f_{w,b}(x^{(i)}) - y^{(i)}\right)$$

α is the **learning rate**.

How does the value of α influence the optimisation algorithm?

- Small α may lead to very long computation times
- Large α may **overshoot** the minimum
- Potential for the optimisation to NOT converge.

Live coding session