Regression

CISC 7026: Introduction to Deep Learning

University of Macau

Participation

Everyone take out paper and a pen

Participation

Everyone take out paper and a pen

Fold the paper into thirds, and write your full name in English and your student ID

Participation

Everyone take out paper and a pen

Fold the paper into thirds, and write your full name in English and your student ID

I will record when you **ask good questions** or **answer questions**

Participation

Everyone take out paper and a pen

Fold the paper into thirds, and write your full name in English and your student ID

I will record when you **ask good questions** or **answer questions**

Participate once: ok participation grade

Participation

Everyone take out paper and a pen

Fold the paper into thirds, and write your full name in English and your student ID

I will record when you **ask good questions** or **answer questions**

Participate once: ok participation grade

Participate more than once: perfect participation grade

Agenda

1. Review

Agenda

1. Review
2. Quiz

Agenda

1. Review
2. Quiz
3. Linear Regression

Agenda

1. **Review**
2. Quiz
3. Linear Regression

Review

We often know **what** we want, but we do not know **how**

Review

We often know **what** we want, but we do not know **how**

We have many pictures of either dogs or muffins $x \in X$

Review

We often know **what** we want, but we do not know **how**

We have many pictures of either dogs or muffins $x \in X$

We want to know if the picture is [dog | muffin] $y \in Y$

Review

We often know **what** we want, but we do not know **how**

We have many pictures of either dogs or muffins $x \in X$

We want to know if the picture is [dog | muffin] $y \in Y$

We learn a function or mapping from X to Y

Review

Why do we call it machine **learning**?

Review

Why do we call it machine **learning**?

We learn the function f from the **data** $x \in X, y \in Y$

Review

Why do we call it machine **learning**?

We learn the function f from the **data** $x \in X, y \in Y$

More specifically, we learn function **parameters** Θ

Review

Why do we call it machine **learning**?

We learn the function f from the **data** $x \in X, y \in Y$

More specifically, we learn function **parameters** Θ

$$f : X \times \Theta \mapsto Y$$

Review

Why do we call it machine **learning**?

We learn the function f from the **data** $x \in X, y \in Y$

More specifically, we learn function **parameters** Θ

$$f : X \times \Theta \mapsto Y$$

$$f\left(你好吗, \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \end{bmatrix} \right) = \text{You good?}$$

Review

Why do we call it machine **learning**?

We learn the function f from the **data** $x \in X, y \in Y$

More specifically, we learn function **parameters** Θ

$$f : X \times \Theta \mapsto Y$$

$$f\left(\text{你好吗}, \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \end{bmatrix}\right) = \text{You good?}$$

$$x = \text{你好吗}, \quad X = \text{Chinese sentences}$$

Review

Why do we call it machine **learning**?

We learn the function f from the **data** $x \in X, y \in Y$

More specifically, we learn function **parameters** Θ

$$f : X \times \Theta \mapsto Y$$

$$f\left(你好吗, \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \end{bmatrix}\right) = \text{You good?}$$

$$x = 你好吗, \quad X = \text{Chinese sentences}$$

$$y = \text{You good?}, \quad Y = \text{English sentences}$$

Review

Create vectors, matrices, or tensors in `jax`

```
import jax.numpy as jnp
a = jnp.array(1) # Scalar
b = jnp.array([1, 2]) # Vector
C = jnp.array([[1,2], [3,4]]) # 2x2 Matrix
D = jnp.ones((3,3,3)) # 3x3x3 Tensor
```

You can determine the dimensions of a variable using `shape`

```
b.shape # Prints (2,)
C.shape # Prints (2,2)
D.shape # prints (3,3,3)
```

Review

Create vectors, matrices, or tensors in `pytorch`

```
import torch
a = torch.tensor(1) # Scalar
b = torch.tensor([1, 2]) # Vector
C = torch.tensor([[1,2], [3,4]]) # 2x2 Matrix
D = torch.ones((3,3,3)) # 3x3x3 Tensor
```

You can determine the dimensions of a variable using `shape`

```
b.shape # Prints (2,)
C.shape # Prints (2,2)
D.shape # prints (3,3,3)
```

Review

```python
import jax.numpy as jnp

s = 5 * jnp.array([1, 2])
print(s) # jnp.array(5, 10)
x = jnp.array([1, 2]) + jnp.array([3, 4])
print(x) # jnp.array([4, 6])
y = jnp.array([1, 2]) * jnp.array([3, 4]) # Careful!
print(y) # jnp.array([3, 8])
z = jnp.array([[1], [2]]) @ jnp.array([[3, 4]])
print(z) # A^t B (dot product), jnp.array([[11]])
```

Review

`pytorch` is very similar to `jax`

```
import torch

s = 5 * torch.tensor([1, 2])
print(s) # torch.tensor(5, 10)
x = torch.tensor([1, 2]) + torch.tensor([3, 4])
print(x) # torch.tensor([4, 6])
y = torch.tensor([1, 2]) * torch.tensor([3, 4]) # Careful!
print(y) # torch.tensor([3, 8])
z = torch.tensor([[1], [2]]) @ torch.tensor([[3, 4]])
print(z) # A^t B (dot product), torch.tensor([[11]])
```

Review

You can also call various methods on arrays/tensors

```
import jax.numpy as jnp

x = jnp.array([[1, 2], [3, 4]]).sum(axis=0)
print(x) # Sum across leading axis, array([4, 6])
y = jnp.array([[1, 2], [3, 4]]).mean()
print(y) # Mean across all axes, array(2.5)
z = jnp.array([[1, 2], [3, 4]]).reshape((4,))
print(z) # jnp.array([1, 2, 3, 4])
```

Review

Same thing for `pytorch`

```
import torch

x = torch.tensor([[1, 2], [3, 4]]).sum(axis=0)
print(x) # Sum across leading axis, array([4, 6])
y = torch.tensor([[1, 2], [3, 4]]).mean()
print(y) # Mean across all axes, array(2.5)
z = torch.tensor([[1, 2], [3, 4]]).reshape((4,))
print(z) # torch.tensor([1, 2, 3, 4])
```

Agenda

1. **Review**
2. Quiz
3. Linear Regression

Agenda

1. Review
2. **Quiz**
3. Linear Regression

Quiz

Time for a quiz!

Quiz

Time for a quiz!

All laptops and phones away

Quiz

Time for a quiz!

All laptops and phones away

Everyone take out paper and pen, write your name and student ID

Quiz

Time for a quiz!

All laptops and phones away

Everyone take out paper and pen, write your name and student ID

I will explain the questions, then you have **15 minutes** to answer the questions

Quiz

Time for a quiz!

All laptops and phones away

Everyone take out paper and pen, write your name and student ID

I will explain the questions, then you have **15 minutes** to answer the questions

You are not expected to answer all questions correctly, do not stress!

Quiz

Time for a quiz!

All laptops and phones away

Everyone take out paper and pen, write your name and student ID

I will explain the questions, then you have **15 minutes** to answer the questions

You are not expected to answer all questions correctly, do not stress!

Quiz

Q1: What is the function signature for machine learning?

$$\_\_ : \_\_\_\_\_ \mapsto \_\_$$

Q2: What does the following code print?

```
jnp.array([1, 2]) * jnp.array([2, 1]) + jnp.array([5, 6])
```

Q3: What does the following code print?

```
jnp.array([[1, 2], [3, 4]]).sum(axis=1)
```

Q4: What does the following code print?

```
torch.arange(4)
```

Quiz

Q1: What is the function signature for machine learning?

Quiz

Q1: What is the function signature for machine learning?

A1: $f : X \times \Theta \mapsto Y$ $\qquad$ $f : X \times \Theta \mapsto Y$ also ok.

Quiz

Q1: What is the function signature for machine learning?

A1: $f : X \times \Theta \mapsto Y$ $\quad\quad$ $f : X \times \Theta \mapsto Y$ also ok.

Q2: What does the following code print?

```
jnp.array([1, 2]) * jnp.array([2, 1]) + jnp.array([5, 6])
```

Quiz

Q1: What is the function signature for machine learning?

A1: $f : X \times \Theta \mapsto Y$ $\qquad$ $f : X \times \Theta \mapsto Y$ also ok.

Q2: What does the following code print?

```
jnp.array([1, 2]) * jnp.array([2, 1]) + jnp.array([5, 6])
```

```
[1 * 2 + 5, 2 * 1 + 6]
```

Quiz

Q1: What is the function signature for machine learning?

A1: $f : X \times \Theta \mapsto Y$ $\qquad$ $f : X \times \Theta \mapsto Y$ also ok.

Q2: What does the following code print?

```
jnp.array([1, 2]) * jnp.array([2, 1]) + jnp.array([5, 6])
```

```
[1 * 2 + 5, 2 * 1 + 6]
```

A2: `Array([7, 8], dtype=int32)`, $\qquad$ `[7, 8]` also ok

Quiz

Q3: What does the following code print?

```
jnp.array([[1, 2], [3, 4]]).sum(axis=1)
```

Quiz

Q3: What does the following code print?

```
jnp.array([[1, 2], [3, 4]]).sum(axis=1)
```

```
[1 + 2, 3 + 4]
```

Quiz

Q3: What does the following code print?

```
jnp.array([[1, 2], [3, 4]]).sum(axis=1)
```

```
[1 + 2, 3 + 4]
```

A3: `Array([3, 7], dtype=int32)`, `[3, 7]` also ok

Quiz

Q3: What does the following code print?

```
jnp.array([[1, 2], [3, 4]]).sum(axis=1)
```

```
[1 + 2, 3 + 4]
```

A3: `Array([3, 7], dtype=int32),` `[3, 7]` also ok

Q4: What does the following code print?

```
jnp.arange(4)
```

Quiz

Q3: What does the following code print?

```
jnp.array([[1, 2], [3, 4]]).sum(axis=1)
```

```
[1 + 2, 3 + 4]
```

A3: `Array([3, 7], dtype=int32)`, `[3, 7]` also ok

Q4: What does the following code print?

```
jnp.arange(4)
```

A4: `Array([0, 1, 2, 3], dtype=int32)`, `[0, 1, 2, 3]` also ok

Agenda

1. Review
2. **Quiz**
3. Linear Regression

Agenda

1. Review
2. Quiz
3. **Linear Regression**

Linear Regression

Today, we will learn about linear regression

Linear Regression

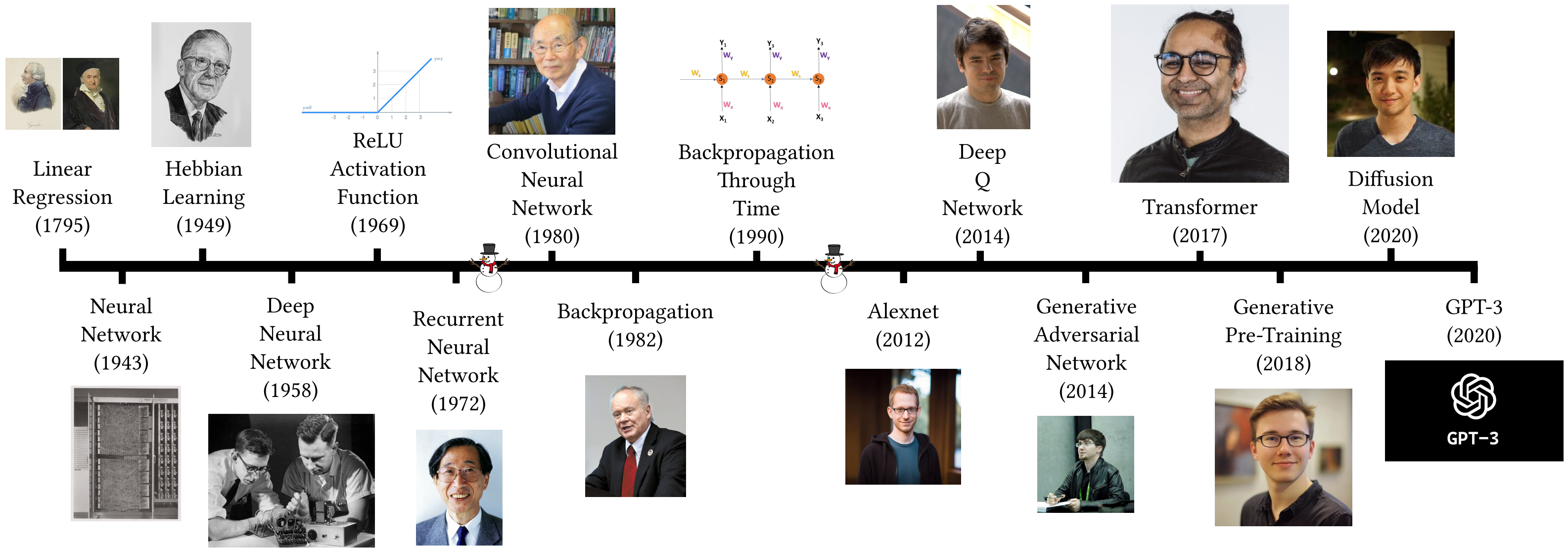
Today, we will learn about linear regression

Probably the oldest method for machine learning (Gauss and Legendre)

Linear Regression

Today, we will learn about linear regression

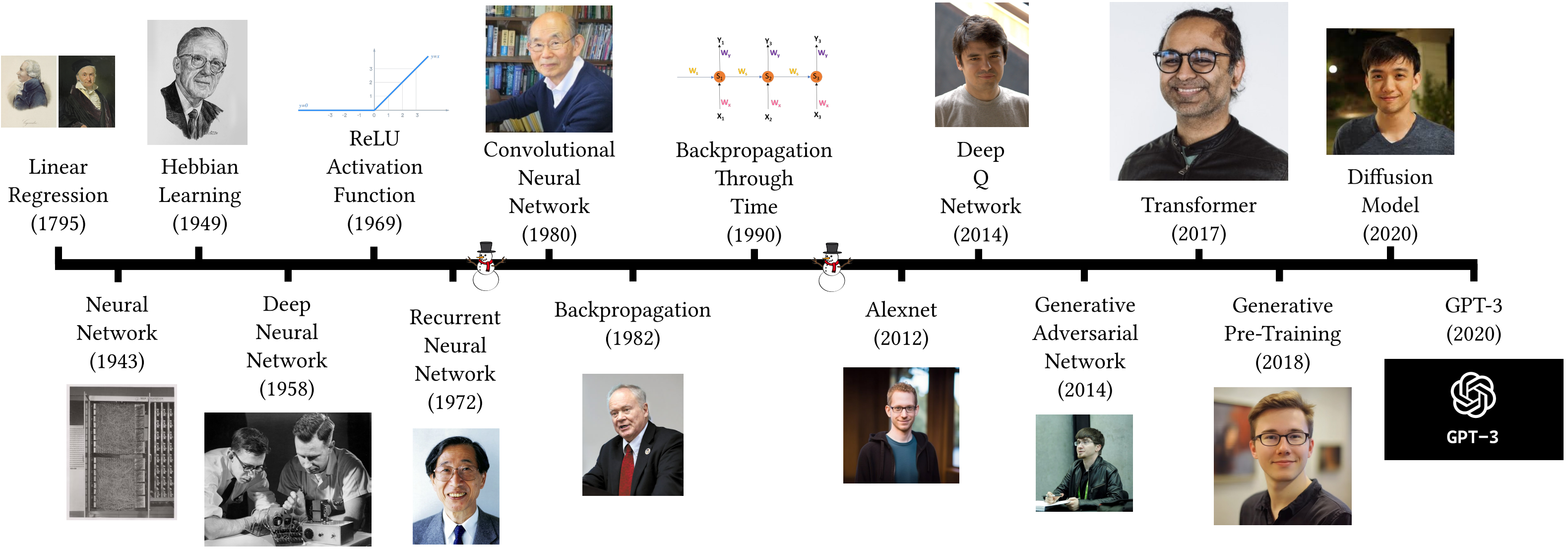
Probably the oldest method for machine learning (Gauss and Legendre)



Linear Regression

Today, we will learn about linear regression

Probably the oldest method for machine learning (Gauss and Legendre)



Neural networks share many similarities with linear regression

Linear Regression

Many problems in ML can be reduced to **regression** or **classification**

Linear Regression

Many problems in ML can be reduced to **regression** or **classification**

Regression asks how many

Linear Regression

Many problems in ML can be reduced to **regression** or **classification**

Regression asks how many

- Given my parents height, How tall will I be?

Linear Regression

Many problems in ML can be reduced to **regression** or **classification**

Regression asks how many
- Given my parents height, How tall will I be?
- Given the rain today, how much rain will there be tomorrow?

Linear Regression

Many problems in ML can be reduced to **regression** or **classification**

Regression asks how many

- Given my parents height, How tall will I be?
- Given the rain today, how much rain will there be tomorrow?
- Given a camera image, how far away is this object?

Linear Regression

Many problems in ML can be reduced to **regression** or **classification**

Regression asks how many

- Given my parents height, How tall will I be?
- Given the rain today, how much rain will there be tomorrow?
- Given a camera image, how far away is this object?

Classification asks which one

Linear Regression

Many problems in ML can be reduced to **regression** or **classification**

Regression asks how many

- Given my parents height, How tall will I be?
- Given the rain today, how much rain will there be tomorrow?
- Given a camera image, how far away is this object?

Classification asks which one

- Is this image of a dog or muffin?

Linear Regression

Many problems in ML can be reduced to **regression** or **classification**

Regression asks how many
- Given my parents height, How tall will I be?
- Given the rain today, how much rain will there be tomorrow?
- Given a camera image, how far away is this object?

Classification asks which one
- Is this image of a dog or muffin?
- Given the rain today, will it rain tomorrow? Yes or no?

Linear Regression

Many problems in ML can be reduced to **regression** or **classification**

Regression asks how many

- Given my parents height, How tall will I be?
- Given the rain today, how much rain will there be tomorrow?
- Given a camera image, how far away is this object?

Classification asks which one

- Is this image of a dog or muffin?
- Given the rain today, will it rain tomorrow? Yes or no?
- Given a camera image, what color is this object? Yellow, blue, red, … ?

Linear Regression

Many problems in ML can be reduced to **regression** or **classification**

Regression asks how many
- Given my parents height, How tall will I be?
- Given the rain today, how much rain will there be tomorrow?
- Given a camera image, how far away is this object?

Classification asks which one
- Is this image of a dog or muffin?
- Given the rain today, will it rain tomorrow? Yes or no?
- Given a camera image, what color is this object? Yellow, blue, red, … ?

Let us start with regression

Linear Regression

Today, we will come up with a regression problem and then solve it!

Linear Regression

Today, we will come up with a regression problem and then solve it!

1. Define an example problem

Linear Regression

Today, we will come up with a regression problem and then solve it!

1. Define an example problem
2. Define our linear model f

Linear Regression

Today, we will come up with a regression problem and then solve it!

1. Define an example problem
2. Define our linear model f
3. Define a loss function $\mathcal{L}$

Linear Regression

Today, we will come up with a regression problem and then solve it!

1. Define an example problem
2. Define our linear model f
3. Define a loss function $\mathcal{L}$
4. Use $\mathcal{L}$ to learn the parameters θ of f

Linear Regression

Today, we will come up with a regression problem and then solve it!

1. Define an example problem
2. Define our linear model f
3. Define a loss function $\mathcal{L}$
4. Use $\mathcal{L}$ to learn the parameters θ of f
5. Solve the example problem
6. Expand to nonlinear models
7. Discuss overfitting
8. Interactive discussion
9. Homework summary

Linear Regression

1. **Define an example problem**
2. Define our linear model f
3. Define a loss function $\mathcal{L}$
4. Use $\mathcal{L}$ to learn the parameters θ of f
5. Solve the example problem
6. Expand to nonlinear models
7. Discuss overfitting
8. Interactive discussion
9. Homework summary

Example Problem

The World Health Organization (WHO) has collected data on life expectancy

Example Problem

The World Health Organization (WHO) has collected data on life expectancy

Available for free at https://www.who.int/data/gho/data/themes/mortality-and-global-health-estimates/ghe-life-expectancy-and-healthy-life-expectancy

Example Problem

The WHO collected data from roughly 3,000 people from 193 countries

Example Problem

The WHO collected data from roughly 3,000 people from 193 countries

For each person, they recorded:

Example Problem

The WHO collected data from roughly 3,000 people from 193 countries

For each person, they recorded:

- Home country

Example Problem

The WHO collected data from roughly 3,000 people from 193 countries

For each person, they recorded:

- Home country
- Alcohol consumption

Example Problem

The WHO collected data from roughly 3,000 people from 193 countries

For each person, they recorded:

- Home country
- Alcohol consumption
- Education

Example Problem

The WHO collected data from roughly 3,000 people from 193 countries

For each person, they recorded:

- Home country
- Alcohol consumption
- Education
- Gross domestic product (GDP) of the country

Example Problem

The WHO collected data from roughly 3,000 people from 193 countries

For each person, they recorded:

- Home country
- Alcohol consumption
- Education
- Gross domestic product (GDP) of the country
- Immunizations for Measles and Hepatitis B

Example Problem

The WHO collected data from roughly 3,000 people from 193 countries

For each person, they recorded:

- Home country
- Alcohol consumption
- Education
- Gross domestic product (GDP) of the country
- Immunizations for Measles and Hepatitis B
- How long this person lived

Example Problem

The WHO collected data from roughly 3,000 people from 193 countries

For each person, they recorded:

- Home country
- Alcohol consumption
- Education
- Gross domestic product (GDP) of the country
- Immunizations for Measles and Hepatitis B
- How long this person lived

We can use this data to make future predictions

Example Problem

Since everyone here is very educated, we will focus on how education affects life expectancy

Example Problem

Since everyone here is very educated, we will focus on how education affects life expectancy

There are studies showing a causal effect of education on health

Example Problem

Since everyone here is very educated, we will focus on how education affects life expectancy

There are studies showing a causal effect of education on health

- *The causal effects of education on health outcomes in the UK Biobank.* Davies et al. *Nature Human Behaviour.*

Example Problem

Since everyone here is very educated, we will focus on how education affects life expectancy

There are studies showing a causal effect of education on health

- *The causal effects of education on health outcomes in the UK Biobank.* Davies et al. *Nature Human Behaviour.*
- By staying in school, you are likely to live longer

Example Problem

Task: Given your education, predict your life expectancy

Example Problem

Task: Given your education, predict your life expectancy

$X \in \mathbb{R}_+$: Years in school

Example Problem

Task: Given your education, predict your life expectancy

$X \in \mathbb{R}_+$: Years in school

$Y \in \mathbb{R}_+$: Age of death

Example Problem

Task: Given your education, predict your life expectancy

$X \in \mathbb{R}_+$: Years in school

$Y \in \mathbb{R}_+$: Age of death

Each $x \in X$ and $y \in Y$ represent a single person

Example Problem

Task: Given your education, predict your life expectancy

$X \in \mathbb{R}_+$: Years in school

$Y \in \mathbb{R}_+$: Age of death

Each $x \in X$ and $y \in Y$ represent a single person

Approach: Learn the parameters θ such that

$$f(x, \theta) = y; \quad x \in X, y \in Y$$

Example Problem

Task: Given your education, predict your life expectancy

$X \in \mathbb{R}_+$: Years in school

$Y \in \mathbb{R}_+$: Age of death

Each $x \in X$ and $y \in Y$ represent a single person

Approach: Learn the parameters θ such that

$$f(x, \theta) = y; \quad x \in X, y \in Y$$

Goal: Given someone's education, predict how long they will live

Linear Regression

1. **Define an example problem**
2. Define our linear model f
3. Define a loss function $\mathcal{L}$
4. Use $\mathcal{L}$ to learn the parameters θ of f
5. Solve the example problem
6. Expand to nonlinear models
7. Discuss overfitting
8. Interactive discussion
9. Homework summary

Linear Regression

1. Define an example problem
2. **Define our linear model f**
3. Define a loss function $\mathcal{L}$
4. Use $\mathcal{L}$ to learn the parameters θ of f
5. Solve the example problem
6. Expand to nonlinear models
7. Discuss overfitting
8. Interactive discussion
9. Homework summary

Linear Model

Soon, f will be a deep neural network

Linear Model

Soon, f will be a deep neural network

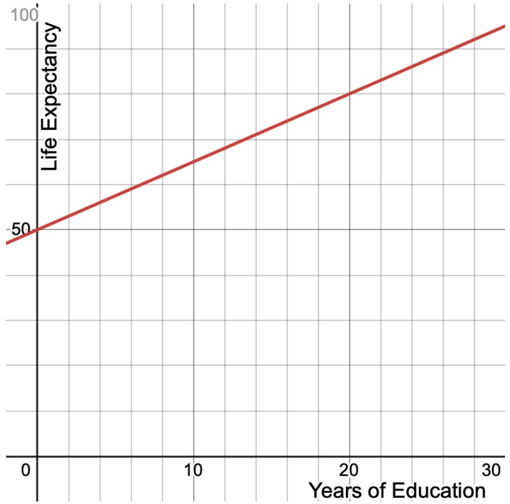
For now, it is easier if we make f a **linear function**

Linear Model

Soon, f will be a deep neural network

For now, it is easier if we make f a **linear function**

$$f(x, \boldsymbol{\theta}) = f\left(x, \begin{bmatrix} \theta_1 \\ \theta_0 \end{bmatrix}\right) = \theta_1 x + \theta_0$$

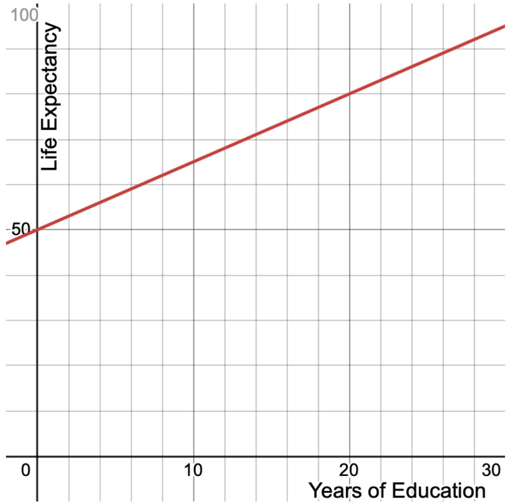


Linear Model

Soon, f will be a deep neural network

For now, it is easier if we make f a **linear function**

$$f(x, \boldsymbol{\theta}) = f\left(x, \begin{bmatrix} \theta_1 \\ \theta_0 \end{bmatrix}\right) = \theta_1 x + \theta_0$$



Now, we need to find the parameters $\boldsymbol{\theta} = \begin{bmatrix} \theta_1 \\ \theta_0 \end{bmatrix}$ that makes $f(x, \boldsymbol{\theta}) = y$

Linear Regression

1. Define an example problem
2. **Define our linear model f**
3. Define a loss function $\mathcal{L}$
4. Use $\mathcal{L}$ to learn the parameters θ of f
5. Solve the example problem
6. Expand to nonlinear models
7. Discuss overfitting
8. Interactive discussion
9. Homework summary

Linear Regression

1. Define an example problem
2. Define our linear model f
3. **Define a loss function $\mathcal{L}$**
4. Use $\mathcal{L}$ to learn the parameters θ of f
5. Solve the example problem
6. Expand to nonlinear models
7. Discuss overfitting
8. Interactive discussion
9. Homework summary

Loss Function

Now, we need to find the parameters $\boldsymbol{\theta} = \begin{bmatrix} \theta_1 \\ \theta_0 \end{bmatrix}$ that make $f(x, \boldsymbol{\theta}) = y$

Loss Function

Now, we need to find the parameters $\boldsymbol{\theta} = \begin{bmatrix} \theta_1 \\ \theta_0 \end{bmatrix}$ that make $f(x, \boldsymbol{\theta}) = y$

How do we find $\boldsymbol{\theta}$? (Hint: We want $f(x, \boldsymbol{\theta}) = y$)

Loss Function

Now, we need to find the parameters $\boldsymbol{\theta} = \begin{bmatrix} \theta_1 \\ \theta_0 \end{bmatrix}$ that make $f(x, \boldsymbol{\theta}) = y$

How do we find $\boldsymbol{\theta}$? (Hint: We want $f(x, \boldsymbol{\theta}) = y$)

We will minimize the **loss** (error) between $f(x, \boldsymbol{\theta})$ and y, for all

$$x \in X, y \in Y$$

Loss Function

We compute the loss using the **loss function**

Loss Function

We compute the loss using the **loss function**

$$\mathcal{L} : X^n \times Y^n \times \Theta \mapsto \mathbb{R}$$

Loss Function

We compute the loss using the **loss function**

$$\mathcal{L} : X^n \times Y^n \times \Theta \mapsto \mathbb{R}$$

The loss function tells us how close $f(x, \boldsymbol{\theta})$ is to y

Loss Function

We compute the loss using the **loss function**

$$\mathcal{L} : X^n \times Y^n \times \Theta \mapsto \mathbb{R}$$

The loss function tells us how close $f(x, \boldsymbol{\theta})$ is to y

By **minimizing** the loss function, we make $f(x, \boldsymbol{\theta}) = y$

Loss Function

We compute the loss using the **loss function**

$$\mathcal{L} : X^n \times Y^n \times \Theta \mapsto \mathbb{R}$$

The loss function tells us how close $f(x, \boldsymbol{\theta})$ is to y

By **minimizing** the loss function, we make $f(x, \boldsymbol{\theta}) = y$

There are many possible loss functions, but for regression we often use the **square error**

Loss Function

We compute the loss using the **loss function**

$$\mathcal{L} : X^n \times Y^n \times \Theta \mapsto \mathbb{R}$$

The loss function tells us how close $f(x, \boldsymbol{\theta})$ is to y

By **minimizing** the loss function, we make $f(x, \boldsymbol{\theta}) = y$

There are many possible loss functions, but for regression we often use the **square error**

$$\text{error}(y, \hat{y}) = (y - \hat{y})^2$$

Loss Function

Let’s derive the error function

Loss Function

Let's derive the error function

$$f(x, \boldsymbol{\theta}) = y$$

f(x) should predict y

Loss Function

Let's derive the error function

$$f(x, \boldsymbol{\theta}) = y$$

f(x) should predict y

$$f(x, \boldsymbol{\theta}) - y = 0$$

Move y to LHS

Loss Function

Let's derive the error function

$$f(x, \boldsymbol{\theta}) = y$$

f(x) should predict y

$$f(x, \boldsymbol{\theta}) - y = 0$$

Move y to LHS

$$(f(x, \boldsymbol{\theta}) - y)^2 = 0$$

Square for minimization

Loss Function

Let's derive the error function

$$f(x, \boldsymbol{\theta}) = y$$

f(x) should predict y

$$f(x, \boldsymbol{\theta}) - y = 0$$

Move y to LHS

$$(f(x, \boldsymbol{\theta}) - y)^2 = 0$$

Square for minimization

$$\text{error}(f(x, \boldsymbol{\theta}), y) = (f(x, \boldsymbol{\theta}) - y)^2$$

Loss Function

We can write the loss function for a single datapoint x_i, y_i as

$$\mathcal{L}(x_i, y_i, \boldsymbol{\theta}) = \text{error}(f(x_i, \boldsymbol{\theta}), y_i) = (f(x_i, \boldsymbol{\theta}) - y_i)^2$$

Loss Function

We can write the loss function for a single datapoint x_i, y_i as

$$\mathcal{L}(x_i, y_i, \boldsymbol{\theta}) = \text{error}(f(x_i, \boldsymbol{\theta}), y_i) = (f(x_i, \boldsymbol{\theta}) - y_i)^2$$

Question: Will this $\mathcal{L}$ give us a good prediction for all possible x?

Loss Function

We can write the loss function for a single datapoint x_i, y_i as

$$\mathcal{L}(x_i, y_i, \boldsymbol{\theta}) = \text{error}(f(x_i, \boldsymbol{\theta}), y_i) = (f(x_i, \boldsymbol{\theta}) - y_i)^2$$

Question: Will this $\mathcal{L}$ give us a good prediction for all possible x?

Answer: No! We only consider a single datapoint x_i, y_i. We want to learn $\boldsymbol{\theta}$ for the entire dataset, for all $x \in X, y \in Y$

Loss Function

For a single x_i, y_i:

$$\mathcal{L}(x_i, y_i, \boldsymbol{\theta}) = \text{error}(f(x_i, \boldsymbol{\theta}), y_i) = (f(x_i, \boldsymbol{\theta}) - y_i)^2$$

Loss Function

For a single x_i, y_i:

$$\mathcal{L}(x_i, y_i, \boldsymbol{\theta}) = \text{error}(f(x_i, \boldsymbol{\theta}), y_i) = (f(x_i, \boldsymbol{\theta}) - y_i)^2$$

For the entire dataset:

$$\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix}^\top, \boldsymbol{y} = \begin{bmatrix} y_1 & y_2 & \dots & y_n \end{bmatrix}^\top$$

Loss Function

For a single x_i, y_i:

$$\mathcal{L}(x_i, y_i, \boldsymbol{\theta}) = \text{error}(f(x_i, \boldsymbol{\theta}), y_i) = (f(x_i, \boldsymbol{\theta}) - y_i)^2$$

For the entire dataset:

$$\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix}^\top, \boldsymbol{y} = \begin{bmatrix} y_1 & y_2 & \dots & y_n \end{bmatrix}^\top$$

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) = \sum_{i=1}^{n} \text{error}(f(x_i, \boldsymbol{\theta}), y_i) = \sum_{i=1}^{n} (f(x_i, \boldsymbol{\theta}) - y_i)^2$$

Loss Function

For a single x_i, y_i:

$$\mathcal{L}(x_i, y_i, \boldsymbol{\theta}) = \text{error}(f(x_i, \boldsymbol{\theta}), y_i) = (f(x_i, \boldsymbol{\theta}) - y_i)^2$$

For the entire dataset:

$$\boldsymbol{x} = [x_1 \ \ x_2 \ \ \dots \ \ x_n]^\top, \boldsymbol{y} = [y_1 \ \ y_2 \ \ \dots \ \ y_n]^\top$$

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) = \sum_{i=1}^{n} \text{error}(f(x_i, \boldsymbol{\theta}), y_i) = \sum_{i=1}^{n} (f(x_i, \boldsymbol{\theta}) - y_i)^2$$

When $\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta})$ is small, then $f(x, \boldsymbol{\theta}) \approx y$ for the whole dataset!

Linear Regression

1. Define an example problem
2. Define our linear model f
3. **Define a loss function $\mathcal{L}$**
4. Use $\mathcal{L}$ to learn the parameters θ of f
5. Solve the example problem
6. Expand to nonlinear models
7. Discuss overfitting
8. Interactive discussion
9. Homework summary

Linear Regression

1. Define an example problem
2. Define our linear model f
3. Define a loss function $\mathcal{L}$
4. **Use $\mathcal{L}$ to learn the parameters θ of f**
5. Solve the example problem
6. Expand to nonlinear models
7. Discuss overfitting
8. Interactive discussion
9. Homework summary

Optimization

Here is our loss function:

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) = \sum_{i=1}^{n} \text{error}(f(x_i, \boldsymbol{\theta}), y_i) = \sum_{i=1}^{n} (f(x_i, \boldsymbol{\theta}) - y_i)^2$$

Optimization

Here is our loss function:

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) = \sum_{i=1}^{n} \text{error}(f(x_i, \boldsymbol{\theta}), y_i) = \sum_{i=1}^{n} (f(x_i, \boldsymbol{\theta}) - y_i)^2$$

When $\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta})$ is small, then $f(x, \boldsymbol{\theta}) \approx y$ for the whole dataset!

Optimization

Here is our loss function:

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) = \sum_{i=1}^{n} \mathrm{error}(f(x_i, \boldsymbol{\theta}), y_i) = \sum_{i=1}^{n} (f(x_i, \boldsymbol{\theta}) - y_i)^2$$

When $\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta})$ is small, then $f(x, \boldsymbol{\theta}) \approx y$ for the whole dataset!

We want to find parameters $\boldsymbol{\theta}$ that make the loss small

Optimization

Here is our loss function:

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) = \sum_{i=1}^{n} \mathrm{error}(f(x_i, \boldsymbol{\theta}), y_i) = \sum_{i=1}^{n} (f(x_i, \boldsymbol{\theta}) - y_i)^2$$

When $\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta})$ is small, then $f(x, \boldsymbol{\theta}) \approx y$ for the whole dataset!

We want to find parameters $\boldsymbol{\theta}$ that make the loss small

Let us state this more formally

Optimization

Our objective is to **minimize** the loss, using $\arg\min$

Optimization

Our objective is to **minimize** the loss, using $\arg\min$

$\arg\min_x f(x)$ means find the x that makes $f(x)$ smallest

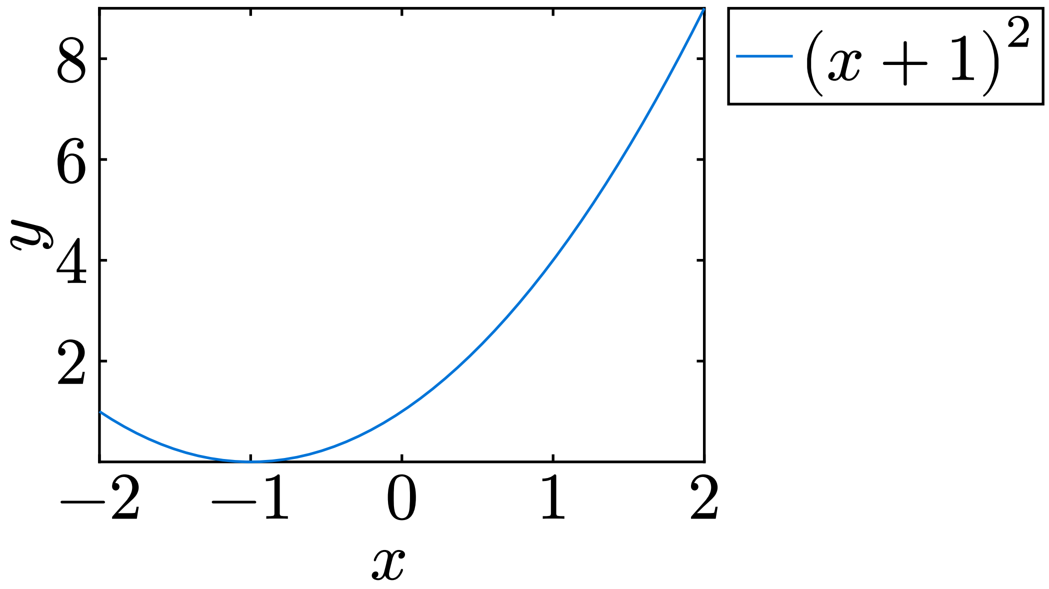
Optimization

Our objective is to **minimize** the loss, using arg min

$\arg\min_x f(x)$ means find the x that makes $f(x)$ smallest

Question:

What is $\arg\min_x (x+1)^2$



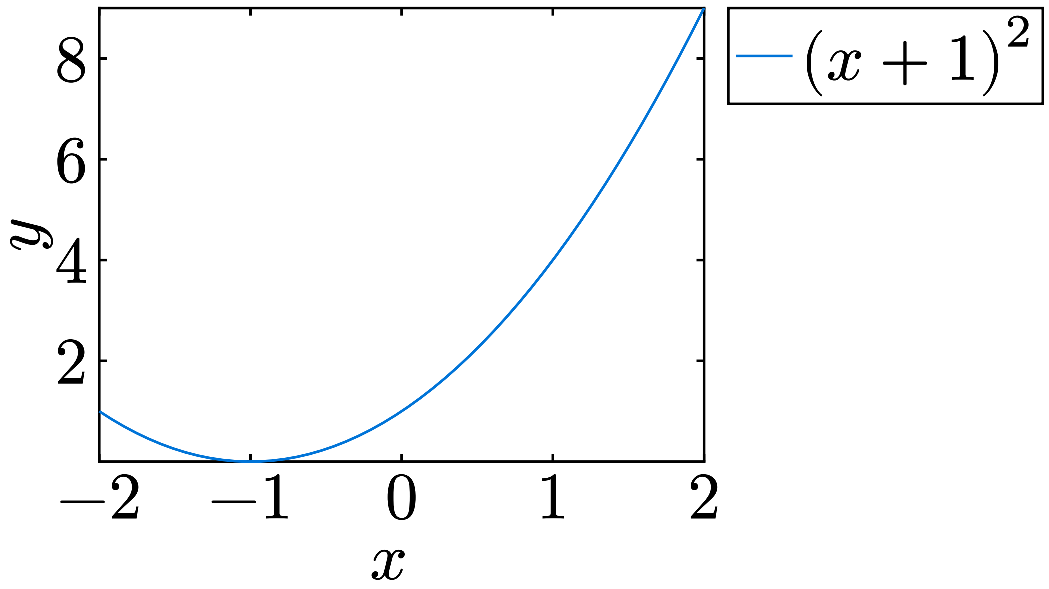
Optimization

Our objective is to **minimize** the loss, using arg min

$\arg\min_x f(x)$ means find the x that makes $f(x)$ smallest

Question:

What is $\underset{x}{\arg\min}\,(x+1)^2$



Answer: $\arg\min_x (x+1)^2 = -1$, where $f(x) = 0$

Optimization

Formally, our objective is to find the arg min of the loss

$$\begin{aligned}\arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) &= \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \text{error}(f(x_i, \boldsymbol{\theta}), y_i) \\ &= \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} (f(x_i, \boldsymbol{\theta}) - y_i)^2\end{aligned}$$

Optimization

$$\arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \text{error}(f(x_i, \boldsymbol{\theta}), y_i)$$

$$= \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} (f(x_i, \boldsymbol{\theta}) - y_i)^2$$

Optimization

$$\arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \text{error}(f(x_i, \boldsymbol{\theta}), y_i)$$

$$= \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} (f(x_i, \boldsymbol{\theta}) - y_i)^2$$

Question: How do we evaluate this expression to find $\boldsymbol{\theta}$?

Optimization

$$\arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \mathrm{error}(f(x_i, \boldsymbol{\theta}), y_i)$$

$$= \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} (f(x_i, \boldsymbol{\theta}) - y_i)^2$$

Question: How do we evaluate this expression to find $\boldsymbol{\theta}$?

Answer: Deriving the solution for this objective requires taking partial derivatives of matrices

Optimization

$$\arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \text{error}(f(x_i, \boldsymbol{\theta}), y_i)$$
$$= \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} (f(x_i, \boldsymbol{\theta}) - y_i)^2$$

Question: How do we evaluate this expression to find $\boldsymbol{\theta}$?

Answer: Deriving the solution for this objective requires taking partial derivatives of matrices

We will derive the solution later. For now, trust me!

Optimization

$$\arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \text{error}(f(x_i, \boldsymbol{\theta}), y_i)$$

$$= \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} (f(x_i, \boldsymbol{\theta}) - y_i)^2$$

Question: How do we evaluate this expression to find $\boldsymbol{\theta}$?

Answer: Deriving the solution for this objective requires taking partial derivatives of matrices

We will derive the solution later. For now, trust me!

We will go over the steps to find $\boldsymbol{\theta}$

Optimization

First, we will construct a **design matrix** $\boldsymbol{X}_D$ containing input data x

Optimization

First, we will construct a **design matrix** $\boldsymbol{X}_D$ containing input data x

$$\boldsymbol{X}_D = [\boldsymbol{x} \;\; \boldsymbol{1}] = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix}$$

Optimization

We add the column of ones so that we can multiply $\boldsymbol{X}_D$ with $\boldsymbol{\theta}$ to get a linear function $\theta_1 x + \theta_0$ evaluated at each data point

$$\boldsymbol{X}_D \boldsymbol{\theta} = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_0 \end{bmatrix} = \underbrace{\begin{bmatrix} \theta_1 x_1 + \theta_0 \\ \theta_1 x_2 + \theta_0 \\ \vdots \\ \theta_1 x_n + \theta_0 \end{bmatrix}}_{\text{Predicted } y}$$

Optimization

With our design matrix $\boldsymbol{X}_D$ and desired output $\boldsymbol{y}$,

$$\boldsymbol{X}_D = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix}, \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

Optimization

With our design matrix $\boldsymbol{X}_D$ and desired output $\boldsymbol{y}$,

$$\boldsymbol{X}_D = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix}, \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

and our parameters $\boldsymbol{\theta}$,

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_1 \\ \theta_0 \end{bmatrix},$$

Optimization

With our design matrix $\boldsymbol{X}_D$ and desired output $\boldsymbol{y}$,

$$\boldsymbol{X}_D = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix}, \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

and our parameters $\boldsymbol{\theta}$,

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_1 \\ \theta_0 \end{bmatrix},$$

$$\boldsymbol{\theta} = \left(\boldsymbol{X}_D^\top \boldsymbol{X}_D\right)^{-1} \boldsymbol{X}_D^\top \boldsymbol{y}$$

(Magic!) We can find the parameters that minimize $\mathcal{L}$

Optimization

To summarize:

Optimization

To summarize:

The $\boldsymbol{\theta}$ given by

$$\boldsymbol{\theta} = \left(\boldsymbol{X}_D^\top \boldsymbol{X}_D \right)^{-1} \boldsymbol{X}_D^\top \boldsymbol{y}$$

Optimization

To summarize:

The $\boldsymbol{\theta}$ given by

$$\boldsymbol{\theta} = \left(\boldsymbol{X}_D^\top \boldsymbol{X}_D\right)^{-1} \boldsymbol{X}_D^\top \boldsymbol{y}$$

Provide the solution to

$$\begin{aligned} \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) &= \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \mathrm{error}(f(x_i, \boldsymbol{\theta}), y_i) \\ &= \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} (f(x_i, \boldsymbol{\theta}) - y_i)^2 \end{aligned}$$

Linear Regression

1. Define an example problem
2. Define our linear model f
3. Define a loss function $\mathcal{L}$
4. **Use $\mathcal{L}$ to learn the parameters θ of f**
5. Solve the example problem
6. Expand to nonlinear models
7. Discuss overfitting
8. Interactive discussion
9. Homework summary

Linear Regression

1. Define an example problem
2. Define our linear model f
3. Define a loss function $\mathcal{L}$
4. Use $\mathcal{L}$ to learn the parameters θ of f
5. **Solve the example problem**
6. Expand to nonlinear models
7. Discuss overfitting
8. Interactive discussion
9. Homework summary

Linear Regression

Back to the example...

Linear Regression

Back to the example...

Task: Given your education, predict your life expectancy

Linear Regression

Back to the example...

Task: Given your education, predict your life expectancy

$X \in \mathbb{R}_+$: Years in school

Linear Regression

Back to the example...

Task: Given your education, predict your life expectancy

$X \in \mathbb{R}_+$: Years in school

$Y \in \mathbb{R}_+$: Age of death

Linear Regression

Back to the example...

Task: Given your education, predict your life expectancy

$X \in \mathbb{R}_+$: Years in school

$Y \in \mathbb{R}_+$: Age of death

Approach: Learn the parameters θ such that

$$f(x, \theta) = y; \quad x \in X, y \in Y$$

Linear Regression

Back to the example...

Task: Given your education, predict your life expectancy

$X \in \mathbb{R}_+$: Years in school

$Y \in \mathbb{R}_+$: Age of death

Approach: Learn the parameters θ such that

$$f(x, \theta) = y; \quad x \in X, y \in Y$$

Goal: Given someone's education, predict how long they will live

Linear Regression

Back to the example…

Task: Given your education, predict your life expectancy

$X \in \mathbb{R}_+$: Years in school

$Y \in \mathbb{R}_+$: Age of death

Approach: Learn the parameters θ such that

$$f(x, \theta) = y; \quad x \in X, y \in Y$$

Goal: Given someone's education, predict how long they will live

You will be doing this in your first assignment!

Linear Regression

Plot the datapoints $(x_1, y_1), (x_2, y_2), ...$

Linear Regression

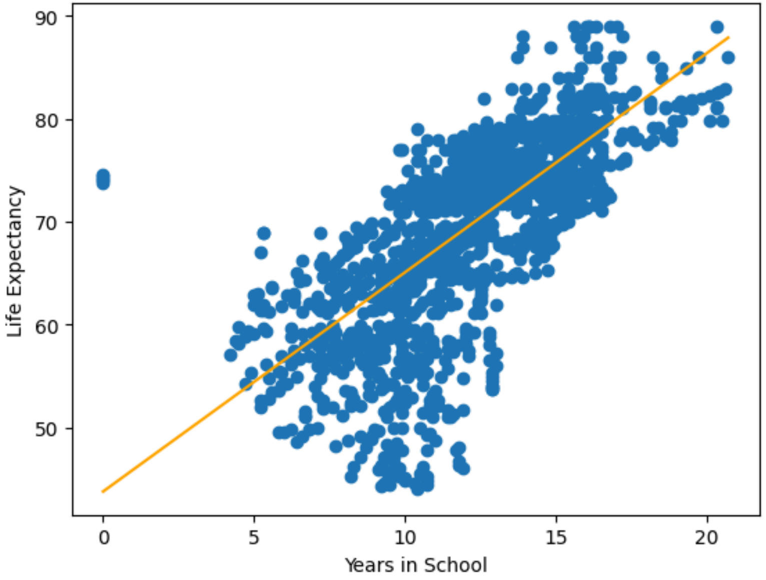
Plot the datapoints $(x_1, y_1), (x_2, y_2), \ldots$

Plot the curve $f(x, \boldsymbol{\theta}) = \theta_1 x + \theta_0; \quad x \in [0, 25]$

Linear Regression

Plot the datapoints $(x_1, y_1), (x_2, y_2), \ldots$

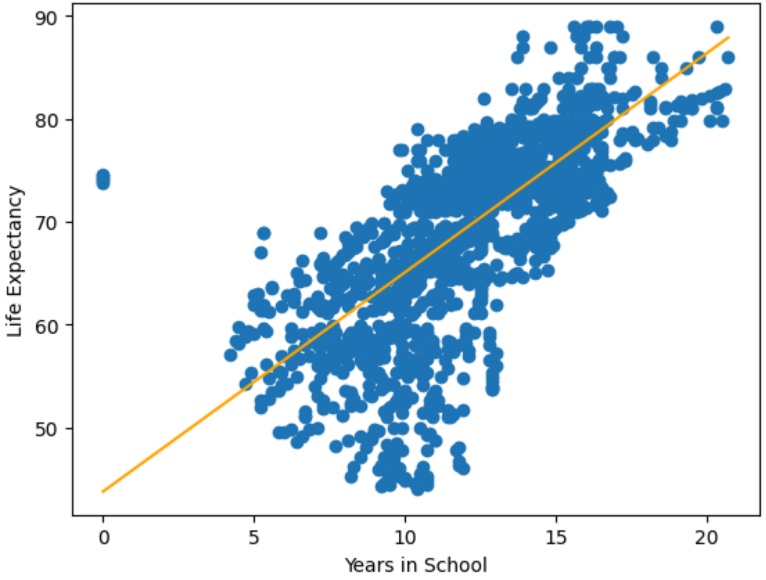
Plot the curve $f(x, \boldsymbol{\theta}) = \theta_1 x + \theta_0; \quad x \in [0, 25]$



Linear Regression

Plot the datapoints $(x_1, y_1), (x_2, y_2), \ldots$

Plot the curve $f(x, \boldsymbol{\theta}) = \theta_1 x + \theta_0; \quad x \in [0, 25]$



Linear Regression

1. Define an example problem
2. Define our linear model f
3. Define a loss function $\mathcal{L}$
4. Use $\mathcal{L}$ to learn the parameters θ of f
5. **Solve the example problem**
6. Expand to nonlinear models
7. Discuss overfitting
8. Interactive discussion
9. Homework summary

Relax

Linear Regression

Task: Given your education, predict your life expectancy

Linear Regression

Task: Given your education, predict your life expectancy

Plot the datapoints $(x_1, y_1), (x_2, y_2), ...$

Linear Regression

Task: Given your education, predict your life expectancy

Plot the datapoints $(x_1, y_1), (x_2, y_2), ...$

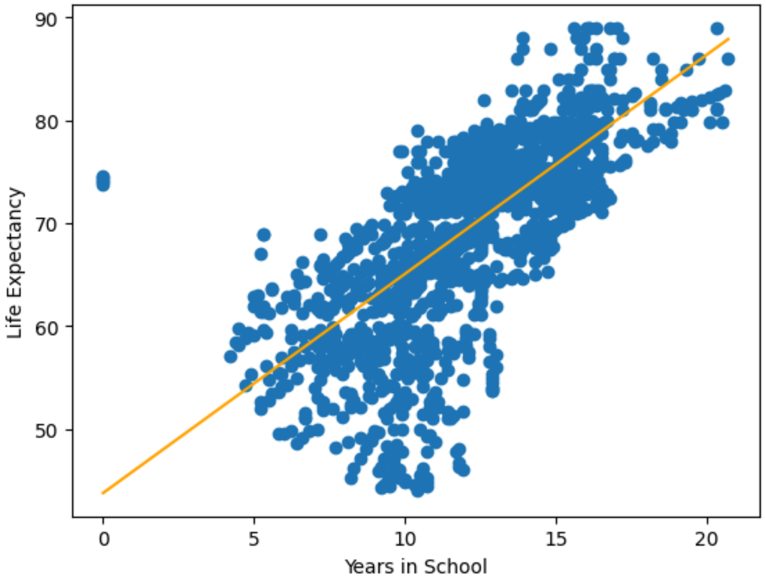
Plot the curve $f(x, \boldsymbol{\theta}) = \theta_1 x + \theta_0; \quad x \in [0, 25]$

Linear Regression

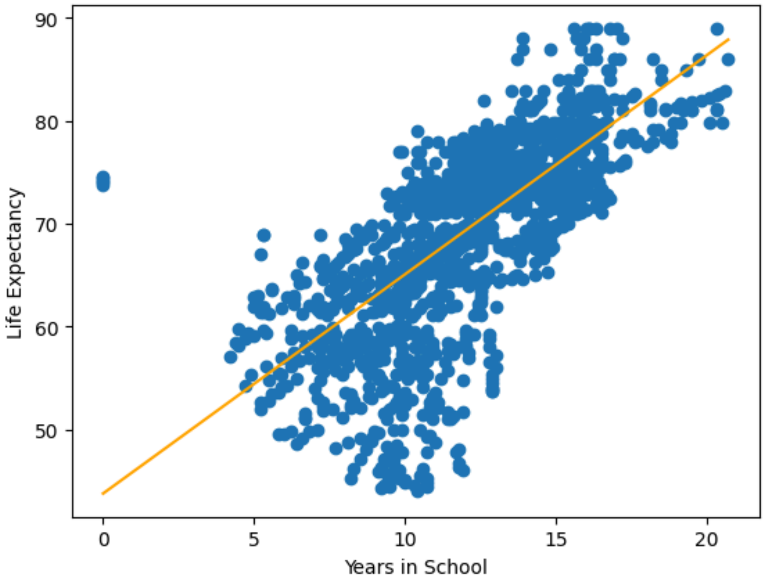
Task: Given your education, predict your life expectancy

Plot the datapoints $(x_1, y_1), (x_2, y_2), \ldots$

Plot the curve $f(x, \boldsymbol{\theta}) = \theta_1 x + \theta_0; \quad x \in [0, 25]$

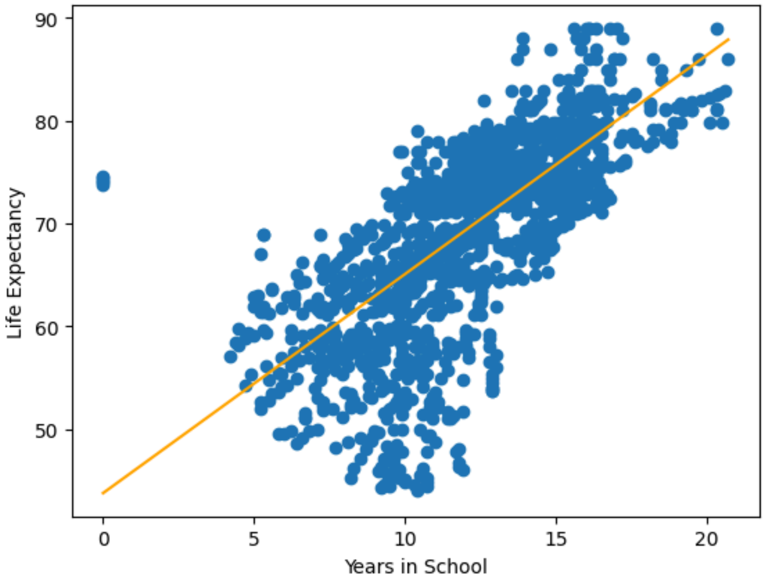


Linear Regression



We figured out linear regression!

Linear Regression



We figured out linear regression!

But can we do better?

Linear Regression

1. Define an example problem
2. Define our linear model f
3. Define a loss function $\mathcal{L}$
4. Use $\mathcal{L}$ to learn the parameters θ of f
5. **Solve the example problem**
6. Expand to nonlinear models
7. Discuss overfitting
8. Interactive discussion
9. Homework summary

Linear Regression

1. Define an example problem
2. Define our linear model f
3. Define a loss function $\mathcal{L}$
4. Use $\mathcal{L}$ to learn the parameters θ of f
5. Solve the example problem
6. **Expand to nonlinear models**
7. Discuss overfitting
8. Interactive discussion
9. Homework summary

Nonlinear Regression

Question:

Nonlinear Regression

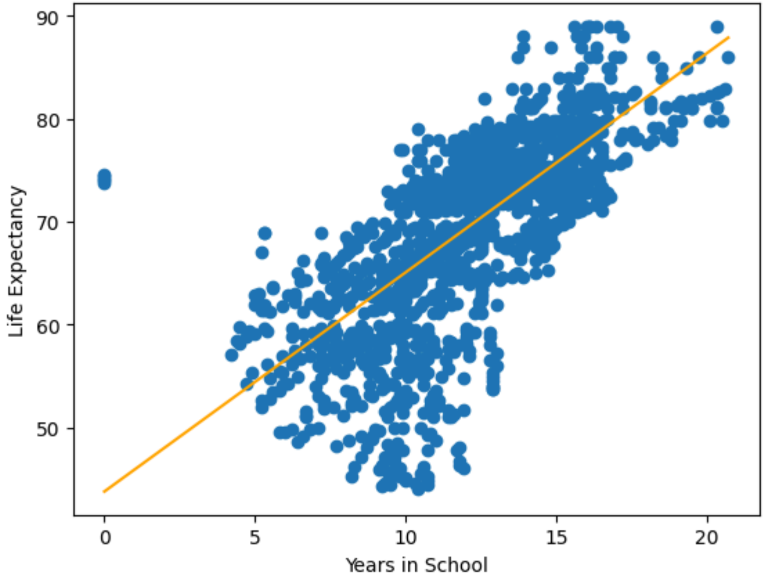
Question:

Does the data look linear?

Nonlinear Regression

Question:

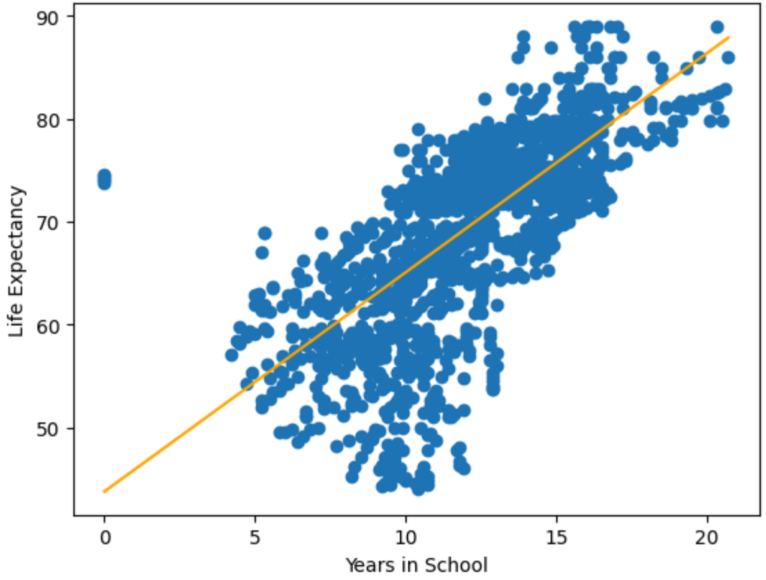
Does the data look linear?



Nonlinear Regression

Question:

Does the data look linear?

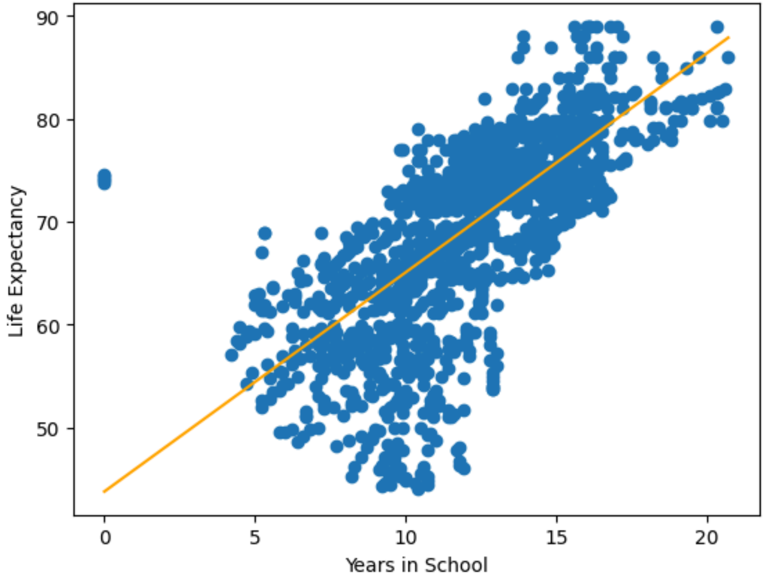


Or maybe more logarithmic?

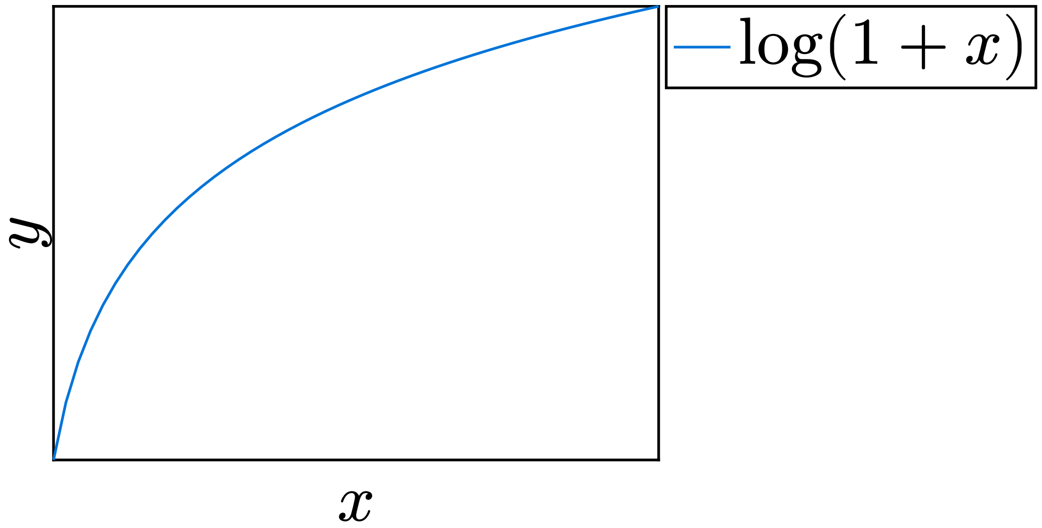
Nonlinear Regression

Question:

Does the data look linear?



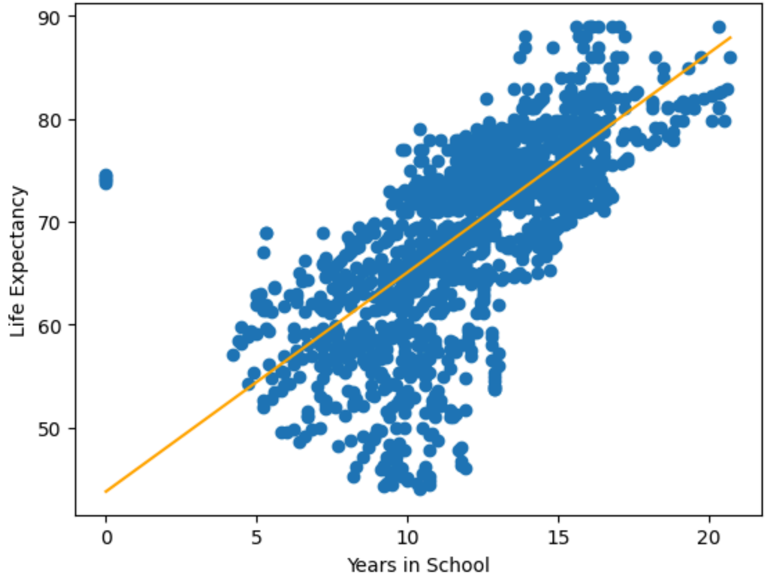
Or maybe more logarithmic?



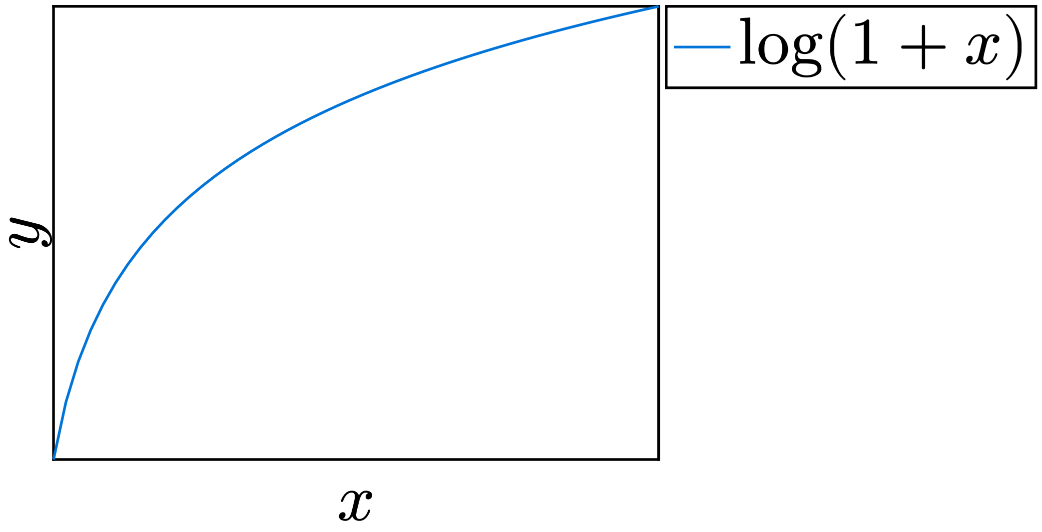
Nonlinear Regression

Question:

Does the data look linear?



Or maybe more logarithmic?



However, linear regression must be linear!

Nonlinear Regression

Question: What does it mean when we say linear regression is linear?

Nonlinear Regression

Question: What does it mean when we say linear regression is linear?

Answer: The function $f(x, \theta)$ is a linear function of x

Nonlinear Regression

Question: What does it mean when we say linear regression is linear?

Answer: The function $f(x, \theta)$ is a linear function of x

Trick: Change of variables to make f nonlinear: $x_{\text{new}} = \log(1 + x_{\text{data}})$

Nonlinear Regression

Question: What does it mean when we say linear regression is linear?

Answer: The function $f(x, \theta)$ is a linear function of x

Trick: Change of variables to make f nonlinear: $x_{\text{new}} = \log(1 + x_{\text{data}})$

$$\boldsymbol{X}_D = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix} \Rightarrow \boldsymbol{X}_D = \begin{bmatrix} \log(1 + x_1) & 1 \\ \log(1 + x_2) & 1 \\ \vdots & \vdots \\ \log(1 + x_n) & 1 \end{bmatrix}$$

Now, f is a linear function of $\log(1 + x)$ – a nonlinear function of x!

Nonlinear Regression

New design matrix...

$$\boldsymbol{X}_D = \begin{bmatrix} \log(1 + x_1) & 1 \\ \log(1 + x_2) & 1 \\ \vdots & \vdots \\ \log(1 + x_n) & 1 \end{bmatrix}$$

New function...

$$f\left(x, \begin{bmatrix} \theta_1 \\ \theta_0 \end{bmatrix}\right) = \theta_1 \log(1 + x) + \theta_0$$

Nonlinear Regression

New design matrix...

$$\boldsymbol{X}_D = \begin{bmatrix} \log(1 + x_1) & 1 \\ \log(1 + x_2) & 1 \\ \vdots & \vdots \\ \log(1 + x_n) & 1 \end{bmatrix}$$

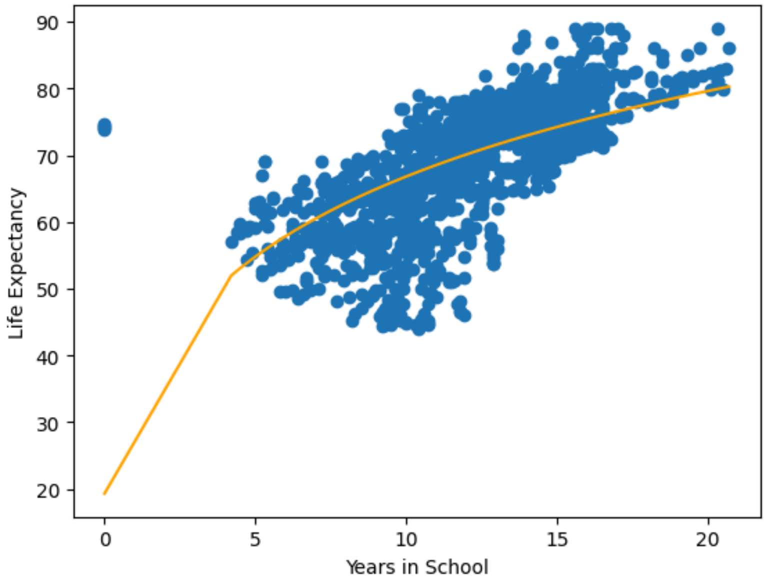
New function...

$$f\left(x, \begin{bmatrix} \theta_1 \\ \theta_0 \end{bmatrix}\right) = \theta_1 \log(1 + x) + \theta_0$$

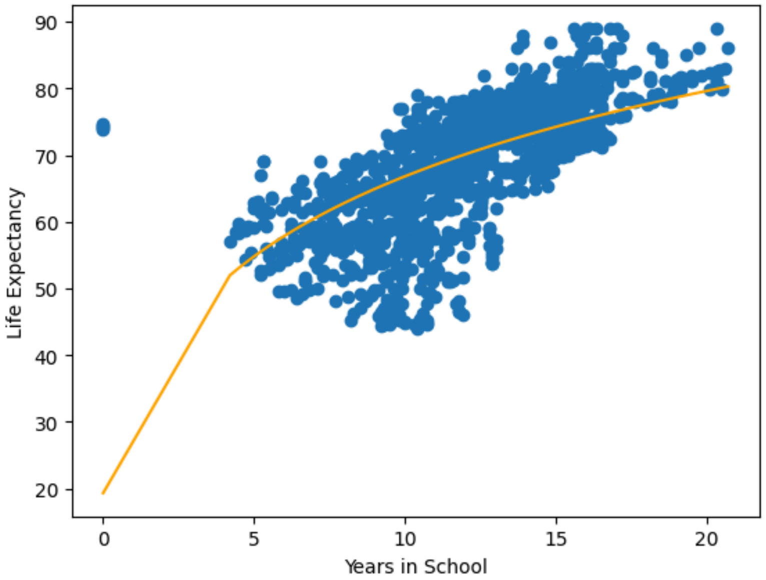
Same solution...

$$\boldsymbol{\theta} = \left(\boldsymbol{X}_D^\top \boldsymbol{X}_D\right)^{-1} \boldsymbol{X}_D^\top \boldsymbol{y}$$

Nonlinear Regression

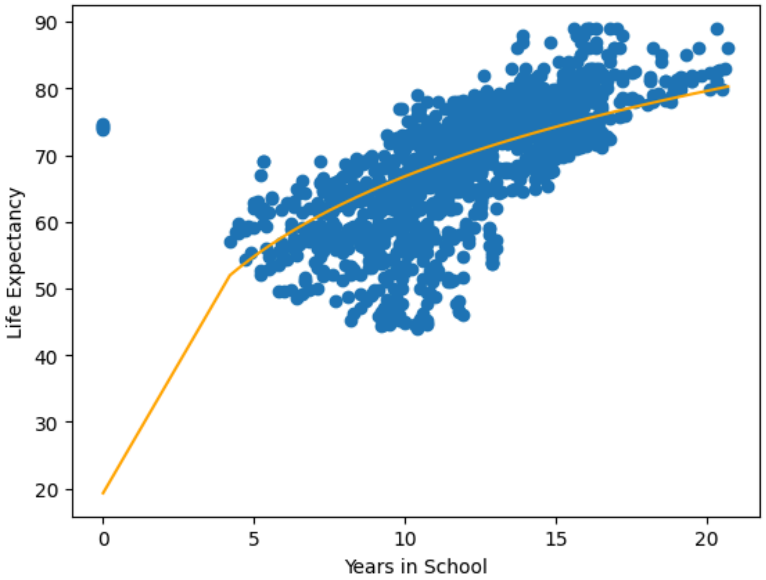


Nonlinear Regression



Better, but still not perfect

Nonlinear Regression



Better, but still not perfect

Can we do even better?

Nonlinear Regression

What about polynomials?

Nonlinear Regression

What about polynomials?

$$f(x) = ax^n + bx^{n-1} + ... + cx + d$$

Nonlinear Regression

What about polynomials?

$$f(x) = ax^n + bx^{n-1} + ... + cx + d$$

Polynomials can approximate **any** function (universal function approximator)

Nonlinear Regression

What about polynomials?

$$f(x) = ax^n + bx^{n-1} + ... + cx + d$$

Polynomials can approximate **any** function (universal function approximator)

Can we extend linear regression to polynomials?

Nonlinear Regression

Expand x to a multi-dimensional input space...

Nonlinear Regression

Expand x to a multi-dimensional input space...

$$\boldsymbol{X}_D = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix} \Rightarrow \boldsymbol{X}_D = \begin{bmatrix} x_1^n & x_1^{n-1} & \dots & x_1 & 1 \\ x_2^n & x_2^{n-1} & \dots & x_2 & 1 \\ \vdots & \vdots & \ddots & & \\ x_n & x_n^{n-1} & \dots & x_n & 1 \end{bmatrix}$$

Nonlinear Regression

Expand x to a multi-dimensional input space...

$$\boldsymbol{X}_D = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix} \Rightarrow \boldsymbol{X}_D = \begin{bmatrix} x_1^n & x_1^{n-1} & \dots & x_1 & 1 \\ x_2^n & x_2^{n-1} & \dots & x_2 & 1 \\ \vdots & \vdots & \ddots & & \\ x_n & x_n^{n-1} & \dots & x_n & 1 \end{bmatrix}$$

And add some new parameters...

$$\boldsymbol{\theta} = [\theta_1 \;\; \theta_0]^\top \Rightarrow \boldsymbol{\theta} = [\theta_n \;\; \theta_{n-1} \;\; \dots \;\; \theta_1 \;\; \theta_0]^\top$$

Nonlinear Regression

$$\boldsymbol{X}_D \boldsymbol{\theta} = \begin{bmatrix} x_1^n & x_1^{n-1} & \dots & x_1 & 1 \\ x_2^n & x_2^{n-1} & \dots & x_2 & 1 \\ \vdots & \vdots & \ddots & & \\ x_n & x_n^{n-1} & \dots & x_n & 1 \end{bmatrix} \begin{bmatrix} \theta_n \\ \theta_{n-1} \\ \vdots \\ \theta_0 \end{bmatrix} = \underbrace{\begin{bmatrix} \theta_n x_1^n + \theta_{n-1} x_1^{n-1} + \dots + \theta_0 \\ \theta_n x_2 + \theta_{n-1} x_2^{n-1} + \dots + \theta_0 \\ \vdots \\ \theta_n x_n^n + \theta_{n-1} x_n^{n-1} + \dots + \theta_0 \end{bmatrix}}_{\text{Y prediction}}$$

Nonlinear Regression

$$\boldsymbol{X}_D \boldsymbol{\theta} = \begin{bmatrix} x_1^n & x_1^{n-1} & \dots & x_1 & 1 \\ x_2^n & x_2^{n-1} & \dots & x_2 & 1 \\ \vdots & \vdots & \ddots & & \\ x_n & x_n^{n-1} & \dots & x_n & 1 \end{bmatrix} \begin{bmatrix} \theta_n \\ \theta_{n-1} \\ \vdots \\ \theta_0 \end{bmatrix} = \underbrace{\begin{bmatrix} \theta_n x_1^n + \theta_{n-1} x_1^{n-1} + \dots + \theta_0 \\ \theta_n x_2 + \theta_{n-1} x_2^{n-1} + \dots + \theta_0 \\ \vdots \\ \theta_n x_n^n + \theta_{n-1} x_n^{n-1} + \dots + \theta_0 \end{bmatrix}}_{\text{Y prediction}}$$

New function... $\quad f(x, \boldsymbol{\theta}) = \theta_n x^n + \theta_{n-1} x^{n-1}, \dots, \theta_1 + x^1 + \theta_0$

Nonlinear Regression

$$\boldsymbol{X}_D \boldsymbol{\theta} = \begin{bmatrix} x_1^n & x_1^{n-1} & \dots & x_1 & 1 \\ x_2^n & x_2^{n-1} & \dots & x_2 & 1 \\ \vdots & \vdots & \ddots & & \\ x_n & x_n^{n-1} & \dots & x_n & 1 \end{bmatrix} \begin{bmatrix} \theta_n \\ \theta_{n-1} \\ \vdots \\ \theta_0 \end{bmatrix} = \underbrace{\begin{bmatrix} \theta_n x_1^n + \theta_{n-1} x_1^{n-1} + \dots + \theta_0 \\ \theta_n x_2 + \theta_{n-1} x_2^{n-1} + \dots + \theta_0 \\ \vdots \\ \theta_n x_n^n + \theta_{n-1} x_n^{n-1} + \dots + \theta_0 \end{bmatrix}}_{\text{Y prediction}}$$

New function... $\quad f(x, \boldsymbol{\theta}) = \theta_n x^n + \theta_{n-1} x^{n-1}, \dots, \theta_1 + x^1 + \theta_0$

Same solution... $\quad \boldsymbol{\theta} = \left(\boldsymbol{X}_D^\top \boldsymbol{X}_D \right)^{-1} \boldsymbol{X}_D^\top \boldsymbol{y}$

Nonlinear Regression

$$f(x, \boldsymbol{\theta}) = \theta_n x^n + \theta_{n-1} x^{n-1}, ..., \theta_1 + x^1 + \theta_0$$

Nonlinear Regression

$$f(x, \boldsymbol{\theta}) = \theta_n x^n + \theta_{n-1} x^{n-1}, ..., \theta_1 + x^1 + \theta_0$$

Summary: By changing the input space, we can fit a polynomial to the data using a linear fit!

Linear Regression

1. Define an example problem
2. Define our linear model f
3. Define a loss function $\mathcal{L}$
4. Use $\mathcal{L}$ to learn the parameters θ of f
5. Solve the example problem
6. **Expand to nonlinear models**
7. Discuss overfitting
8. Interactive discussion
9. Homework summary

Linear Regression

1. Define an example problem
2. Define our linear model f
3. Define a loss function $\mathcal{L}$
4. Use $\mathcal{L}$ to learn the parameters θ of f
5. Solve the example problem
6. Expand to nonlinear models
7. **Discuss overfitting**
8. Interactive discussion
9. Homework summary

Overfitting

$$f(x, \boldsymbol{\theta}) = \theta_n x^n + \theta_{n-1} x^{n-1}, ..., \theta_1 + x^1 + \theta_0$$

Overfitting

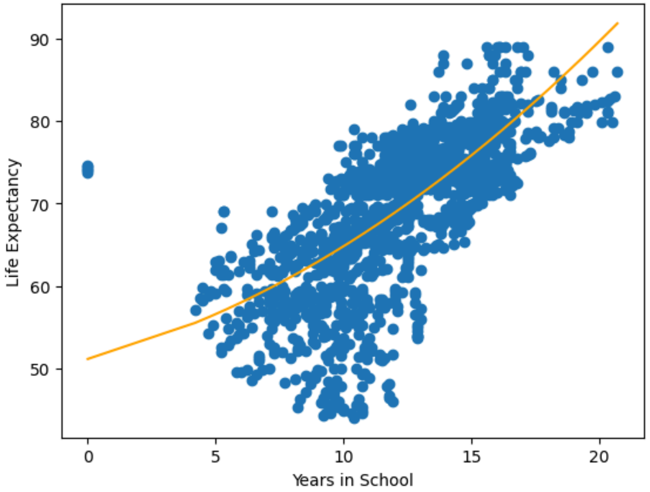
$$f(x, \boldsymbol{\theta}) = \theta_n x^n + \theta_{n-1} x^{n-1}, ..., \theta_1 + x^1 + \theta_0$$

How do we choose n (polynomial order) that provides the best fit?

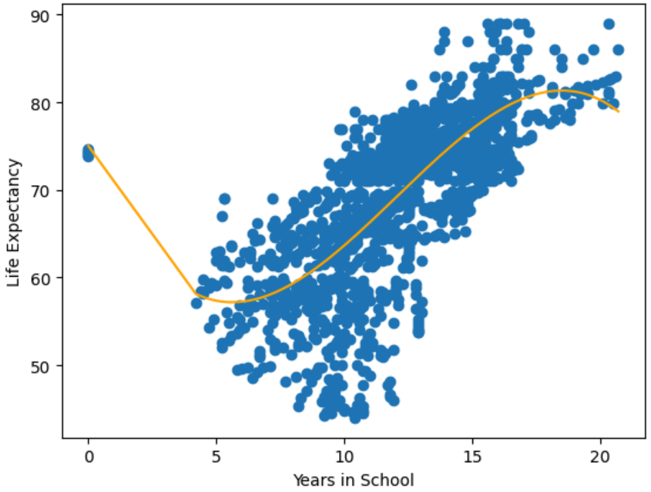
Overfitting

$$f(x, \boldsymbol{\theta}) = \theta_n x^n + \theta_{n-1} x^{n-1}, ..., \theta_1 + x^1 + \theta_0$$

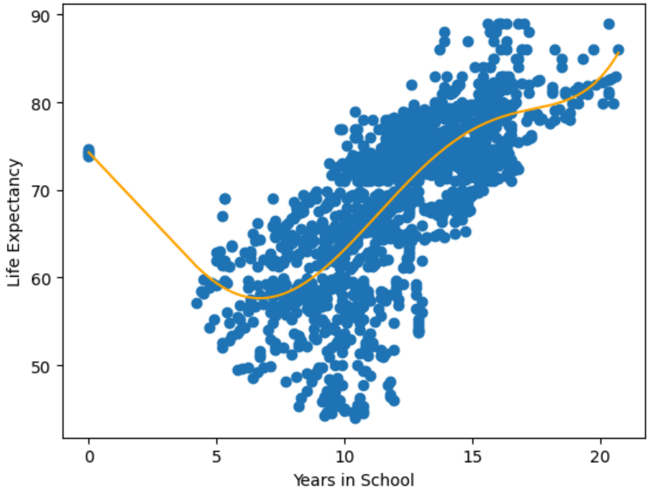
How do we choose n (polynomial order) that provides the best fit?



$n = 2$



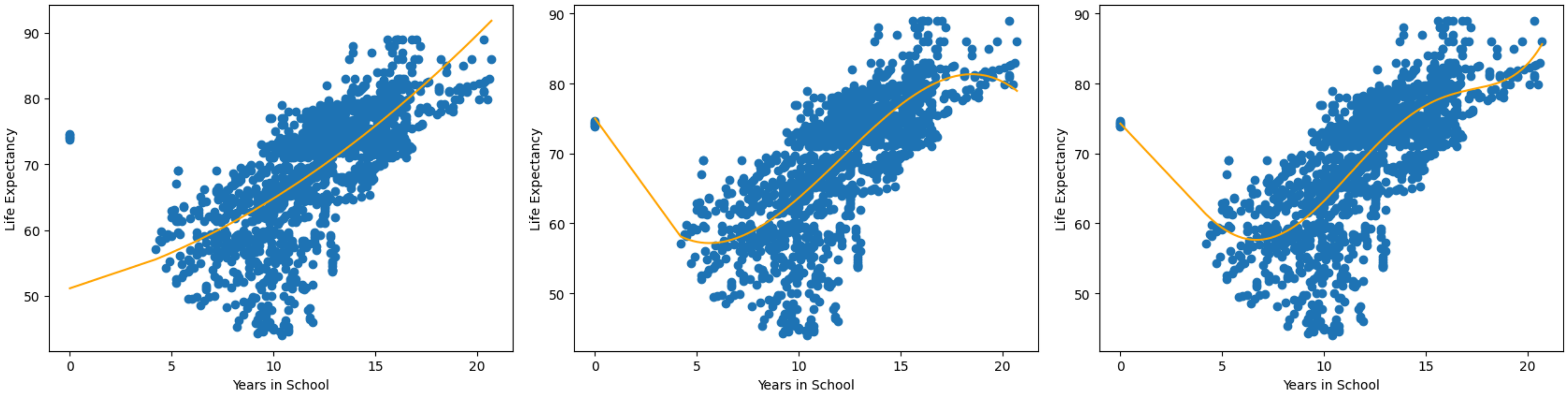
$n = 3$



$n = 5$

Overfitting

How do we choose n (polynomial order) that provides the best fit?

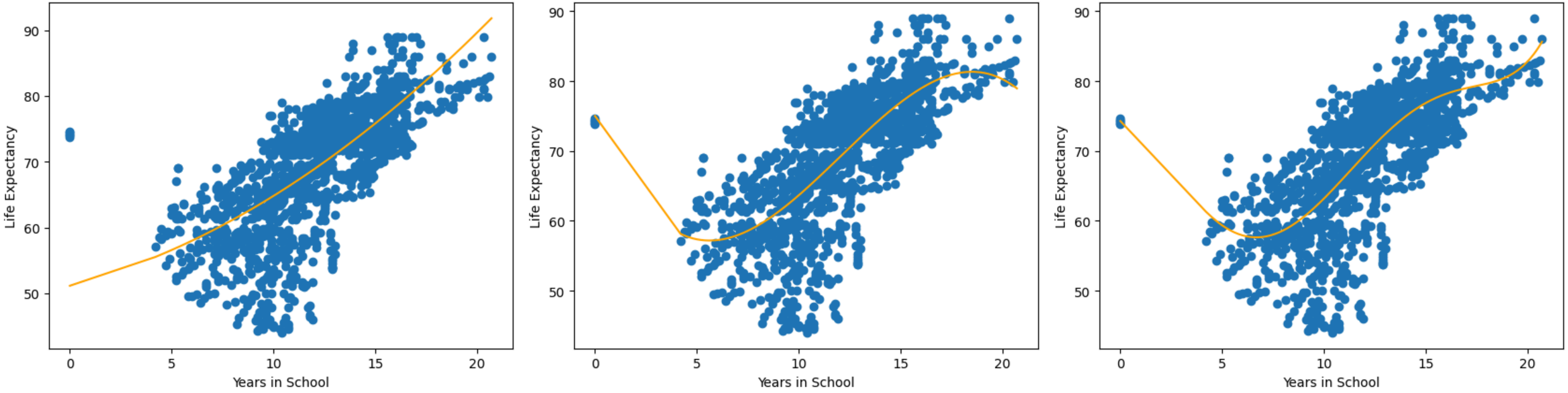


$n = 2$ $\qquad$ $n = 3$ $\qquad$ $n = 5$

Pick the n with the smallest loss

$$\arg\min_{\boldsymbol{\theta}, n} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, (\boldsymbol{\theta}, n))$$

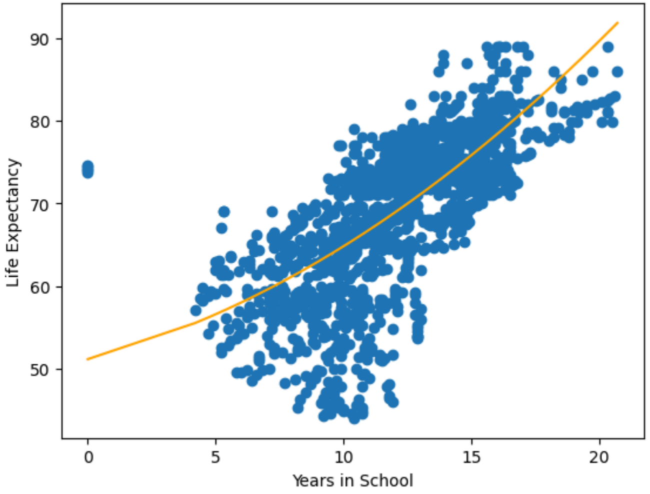
Overfitting

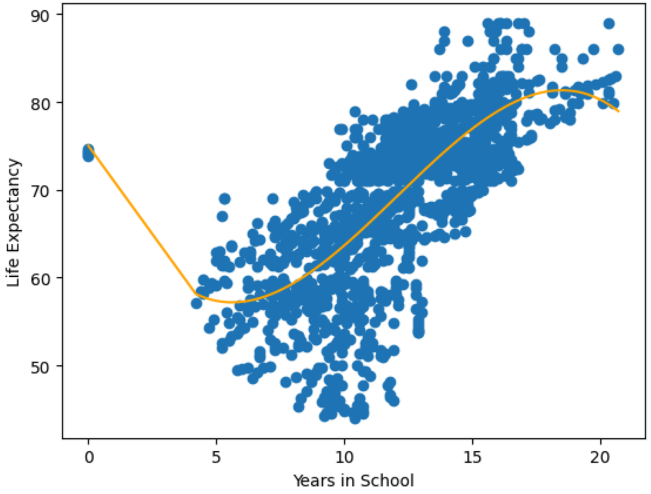


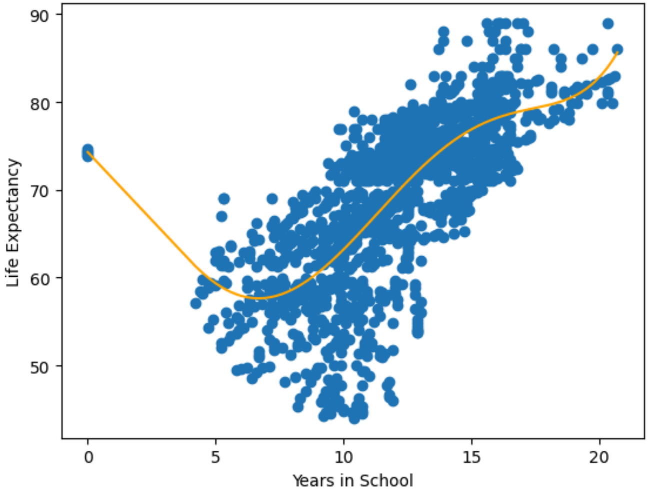
$n = 2$ $n = 3$ $n = 5$

Question: Which n do you think has the smallest loss?

Overfitting





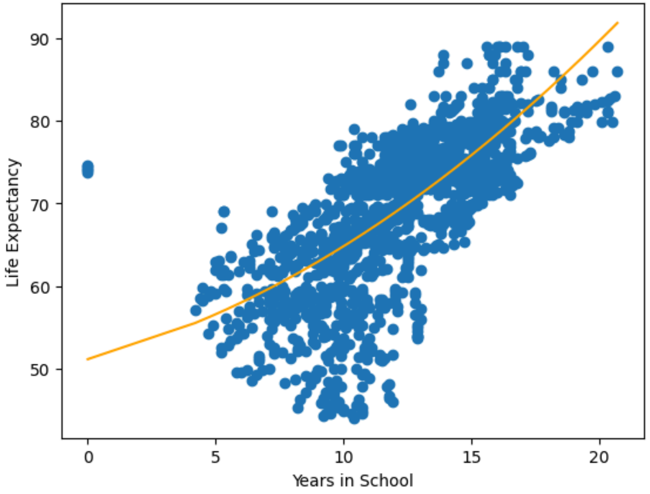


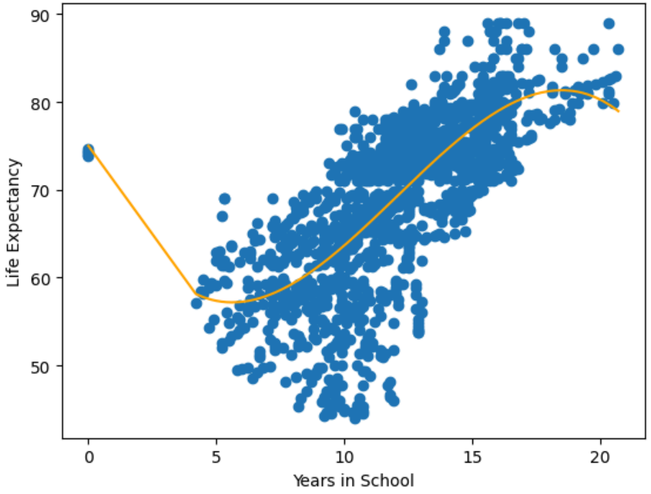
$n = 2$ $\qquad$ $n = 3$ $\qquad$ $n = 5$

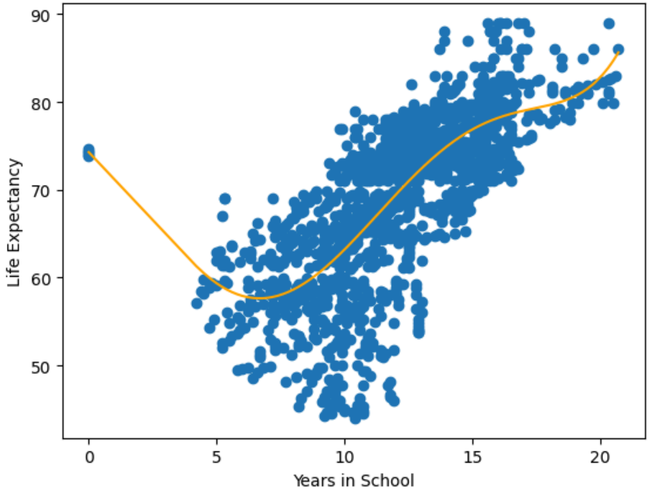
Question: Which n do you think has the smallest loss?

Answer: $n = 5$, but intuitively, $n = 5$ does not seem very good...

Overfitting



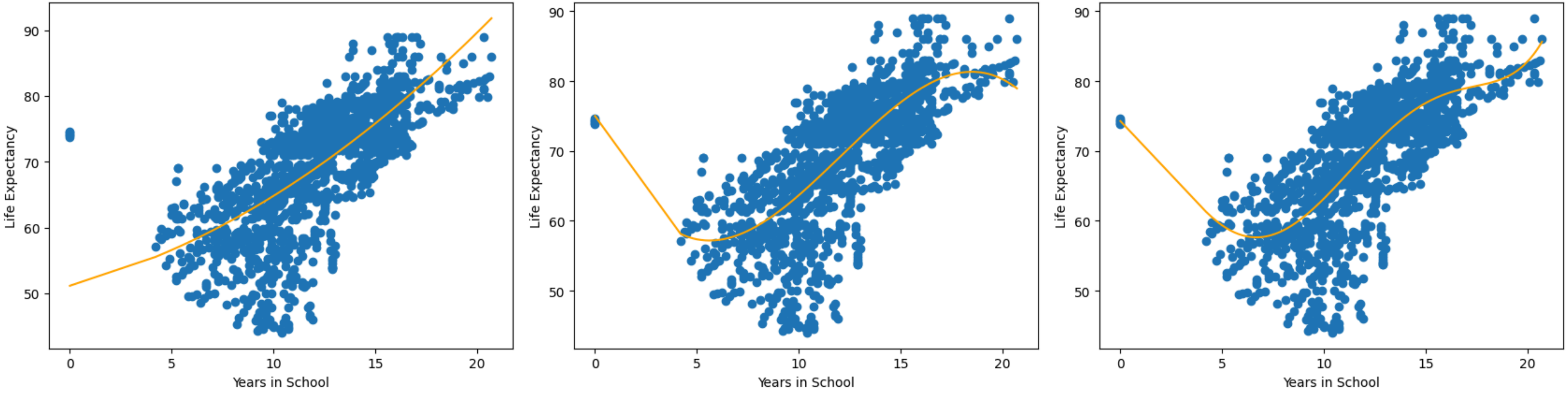




$n = 2$ $\qquad$ $n = 3$ $\qquad$ $n = 5$

More specifically, $n = 5$ will not generalize to new data

Overfitting

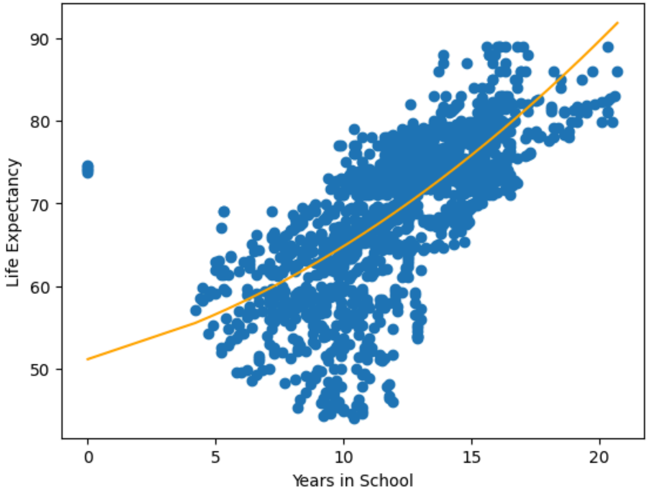


$n = 2$ $\qquad$ $n = 3$ $\qquad$ $n = 5$

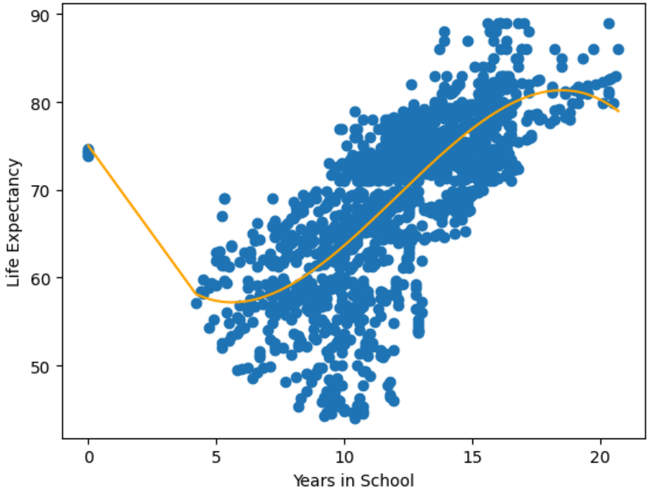
More specifically, $n = 5$ will not generalize to new data

We will only use our model for new data (we already have the y for a known x)!

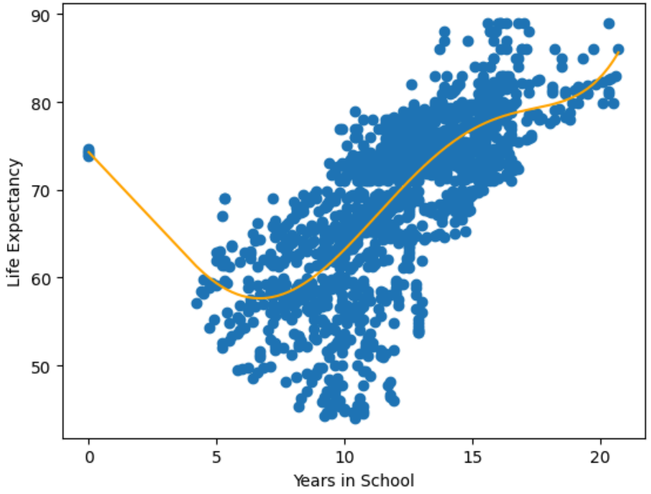
Overfitting



$n = 2$



$n = 3$



$n = 5$

More specifically, $n = 5$ will not generalize to new data

We will only use our model for new data (we already have the y for a known x)!

Overfitting

When our model has a small loss but does not generalize to new data, we call it **overfitting**

Overfitting

When our model has a small loss but does not generalize to new data, we call it **overfitting**

The model has fit too closely to the sampled data points, rather than the trend

Overfitting

When our model has a small loss but does not generalize to new data, we call it **overfitting**

The model has fit too closely to the sampled data points, rather than the trend

Models that overfit are not useful for making predictions

Overfitting

When our model has a small loss but does not generalize to new data, we call it **overfitting**

The model has fit too closely to the sampled data points, rather than the trend

Models that overfit are not useful for making predictions

Back to the question...

Overfitting

When our model has a small loss but does not generalize to new data, we call it **overfitting**

The model has fit too closely to the sampled data points, rather than the trend

Models that overfit are not useful for making predictions

Back to the question...

Question: How do we choose n such that our polynomial model works for unseen/new data?

Overfitting

When our model has a small loss but does not generalize to new data, we call it **overfitting**

The model has fit too closely to the sampled data points, rather than the trend

Models that overfit are not useful for making predictions

Back to the question…

Question: How do we choose n such that our polynomial model works for unseen/new data?

Answer: Compute the loss on unseen data!

Overfitting

To compute the loss on unseen data, we will need unseen data

Overfitting

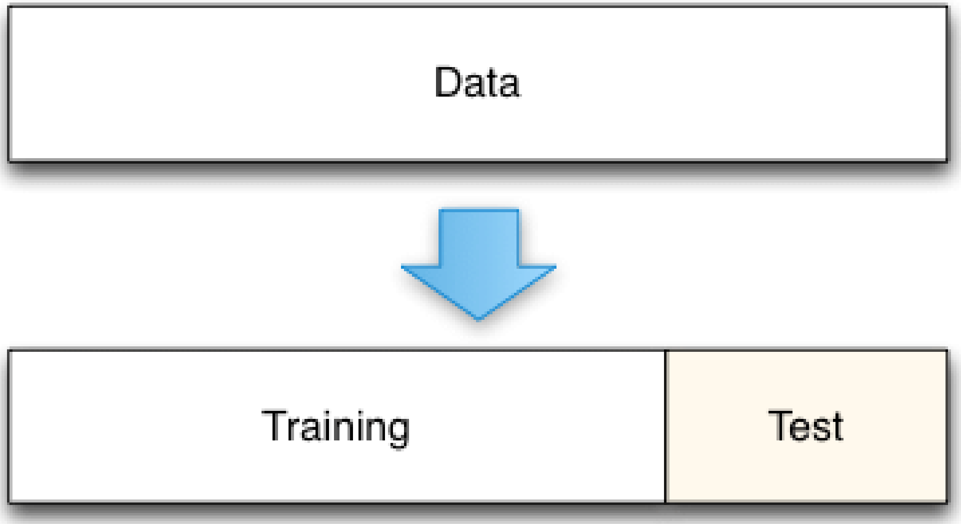
To compute the loss on unseen data, we will need unseen data

Let us create some unseen data!

Overfitting

To compute the loss on unseen data, we will need unseen data

Let us create some unseen data!



Overfitting

Question: How do we choose the training and testing datasets?

Overfitting

Question: How do we choose the training and testing datasets?

$$\text{Option 1: } \boldsymbol{x}_{\text{train}} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \boldsymbol{y}_{\text{train}} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}; \quad \boldsymbol{x}_{\text{test}} = \begin{bmatrix} x_4 \\ x_5 \end{bmatrix}; \boldsymbol{y}_{\text{test}} = \begin{bmatrix} y_4 \\ y_5 \end{bmatrix}$$

Overfitting

Question: How do we choose the training and testing datasets?

$$\text{Option 1: } \boldsymbol{x}_{\text{train}} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \boldsymbol{y}_{\text{train}} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}; \quad \boldsymbol{x}_{\text{test}} = \begin{bmatrix} x_4 \\ x_5 \end{bmatrix}; \boldsymbol{y}_{\text{test}} = \begin{bmatrix} y_4 \\ y_5 \end{bmatrix}$$

$$\text{Option 2: } \boldsymbol{x}_{\text{train}} = \begin{bmatrix} x_4 \\ x_1 \\ x_3 \end{bmatrix} \boldsymbol{y}_{\text{train}} = \begin{bmatrix} y_4 \\ y_1 \\ y_3 \end{bmatrix}; \quad \boldsymbol{x}_{\text{test}} = \begin{bmatrix} x_2 \\ x_5 \end{bmatrix}; \boldsymbol{y}_{\text{test}} = \begin{bmatrix} y_2 \\ y_5 \end{bmatrix}$$

Overfitting

Question: How do we choose the training and testing datasets?

$$\text{Option 1: } \boldsymbol{x}_{\text{train}} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \boldsymbol{y}_{\text{train}} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}; \quad \boldsymbol{x}_{\text{test}} = \begin{bmatrix} x_4 \\ x_5 \end{bmatrix}; \boldsymbol{y}_{\text{test}} = \begin{bmatrix} y_4 \\ y_5 \end{bmatrix}$$

$$\text{Option 2: } \boldsymbol{x}_{\text{train}} = \begin{bmatrix} x_4 \\ x_1 \\ x_3 \end{bmatrix} \boldsymbol{y}_{\text{train}} = \begin{bmatrix} y_4 \\ y_1 \\ y_3 \end{bmatrix}; \quad \boldsymbol{x}_{\text{test}} = \begin{bmatrix} x_2 \\ x_5 \end{bmatrix}; \boldsymbol{y}_{\text{test}} = \begin{bmatrix} y_2 \\ y_5 \end{bmatrix}$$

Answer: Always shuffle the data

Overfitting

Question: How do we choose the training and testing datasets?

$$\text{Option 1: } \boldsymbol{x}_{\text{train}} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \boldsymbol{y}_{\text{train}} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}; \quad \boldsymbol{x}_{\text{test}} = \begin{bmatrix} x_4 \\ x_5 \end{bmatrix}; \boldsymbol{y}_{\text{test}} = \begin{bmatrix} y_4 \\ y_5 \end{bmatrix}$$

$$\text{Option 2: } \boldsymbol{x}_{\text{train}} = \begin{bmatrix} x_4 \\ x_1 \\ x_3 \end{bmatrix} \boldsymbol{y}_{\text{train}} = \begin{bmatrix} y_4 \\ y_1 \\ y_3 \end{bmatrix}; \quad \boldsymbol{x}_{\text{test}} = \begin{bmatrix} x_2 \\ x_5 \end{bmatrix}; \boldsymbol{y}_{\text{test}} = \begin{bmatrix} y_2 \\ y_5 \end{bmatrix}$$

Answer: Always shuffle the data

Note: The model must never see the testing dataset during training. This is very important!

Overfitting

We can now measure how the model generalizes to new data

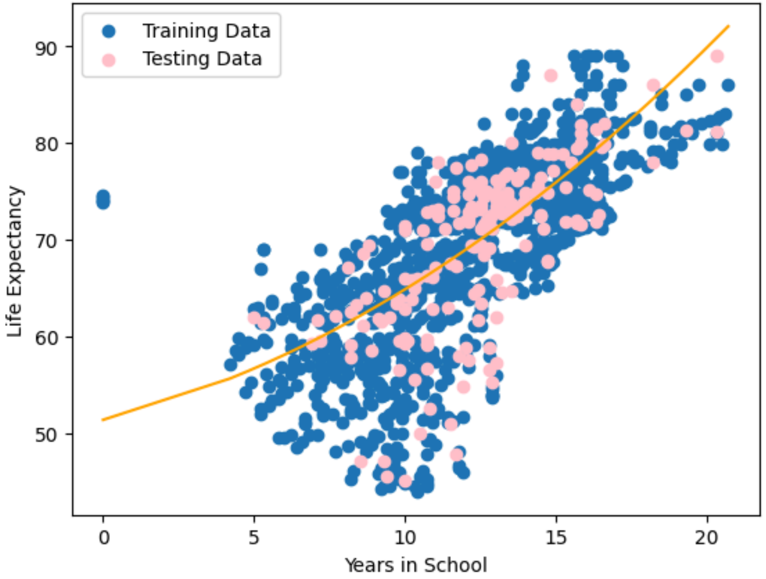
Overfitting

We can now measure how the model generalizes to new data

Learn parameters from the train dataset, evaluate on the test dataset

Overfitting

We can now measure how the model generalizes to new data



Learn parameters from the train dataset, evaluate on the test dataset

$$\mathcal{L}(\boldsymbol{X}_{\text{train}}, \boldsymbol{y}_{\text{train}}, \boldsymbol{\theta}) \qquad\qquad \mathcal{L}(\boldsymbol{X}_{\text{test}}, \boldsymbol{y}_{\text{test}}, \boldsymbol{\theta})$$

Overfitting

We use separate training and testing datasets on **all** machine learning models, not just linear regression

Overfitting

We use separate training and testing datasets on **all** machine learning models, not just linear regression

Linear Regression

1. Define an example problem
2. Define our linear model f
3. Define a loss function $\mathcal{L}$
4. Use $\mathcal{L}$ to learn the parameters θ of f
5. Solve the example problem
6. Expand to nonlinear models
7. **Discuss overfitting**
8. Interactive discussion
9. Homework summary

Linear Regression

1. Define an example problem
2. Define our linear model f
3. Define a loss function $\mathcal{L}$
4. Use $\mathcal{L}$ to learn the parameters θ of f
5. Solve the example problem
6. Expand to nonlinear models
7. Discuss overfitting
8. **Interactive discussion**
9. Homework summary

Discussion

Q: When should we use test and train splits?

Discussion

Q: When should we use test and train splits?

Q: Are neural networks more powerful than linear regression?

Discussion

Q: When should we use test and train splits?

Q: Are neural networks more powerful than linear regression?

Q: Why would we want to use linear regression instead of neural networks?

Discussion

Q: When should we use test and train splits?

Q: Are neural networks more powerful than linear regression?

Q: Why would we want to use linear regression instead of neural networks?

Q: What are interesting problems that we can apply linear regression to?

Discussion

Q: When should we use test and train splits?

Q: Are neural networks more powerful than linear regression?

Q: Why would we want to use linear regression instead of neural networks?

Q: What are interesting problems that we can apply linear regression to?

Q: We use a squared error loss. What effect does this have on outliers?

Discussion

Q: When should we use test and train splits?

Q: Are neural networks more powerful than linear regression?

Q: Why would we want to use linear regression instead of neural networks?

Q: What are interesting problems that we can apply linear regression to?

Q: We use a squared error loss. What effect does this have on outliers?

Linear Regression

1. Define an example problem
2. Define our linear model f
3. Define a loss function $\mathcal{L}$
4. Use $\mathcal{L}$ to learn the parameters θ of f
5. Solve the example problem
6. Expand to nonlinear models
7. Discuss overfitting
8. **Interactive discussion**
9. Homework summary

Linear Regression

1. Define an example problem
2. Define our linear model f
3. Define a loss function $\mathcal{L}$
4. Use $\mathcal{L}$ to learn the parameters θ of f
5. Solve the example problem
6. Expand to nonlinear models
7. Discuss overfitting
8. Interactive discussion
9. **Homework summary**

Homework

Tips for assignment 1

Homework

Tips for assignment 1

```python
def f(theta, design):
  # Linear function
  return design @ theta
```

Homework

Tips for assignment 1

```
def f(theta, design):
  # Linear function
  return design @ theta
```

Not all matrices can be inverted! Ensure the matrices are square and the condition number is low

```
A.shape
cond = jax.numpy.linalg.cond(A)
```

Homework

Tips for assignment 1

```python
def f(theta, design):
  # Linear function
  return design @ theta
```

Not all matrices can be inverted! Ensure the matrices are square and the condition number is low

```python
A.shape
cond = jax.numpy.linalg.cond(A)
```

Everything you need is in the lecture notes

Homework

https://colab.research.google.com/drive/1I6YgapkfaU71RdOotaTPLYdX9WflV1me