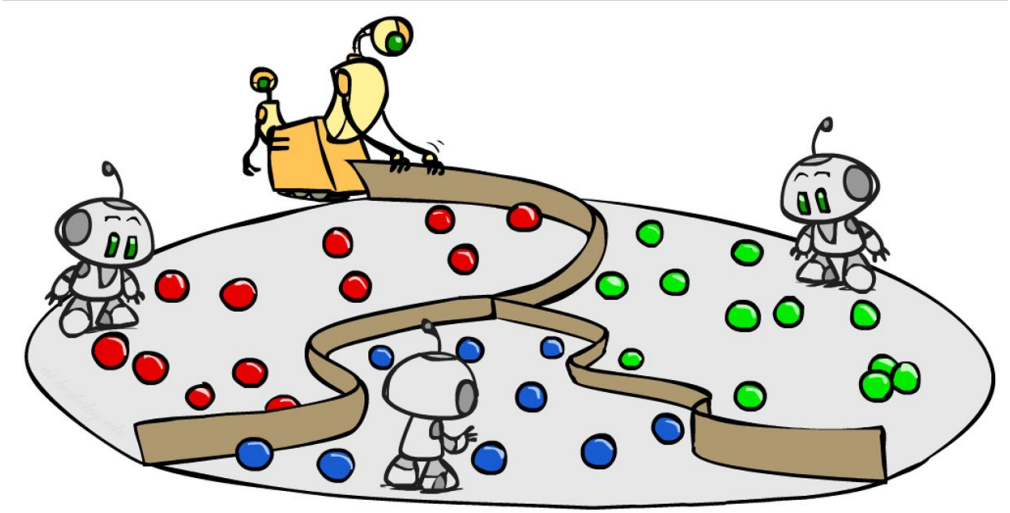
CS-ELEC2C: Machine Learning

Loss Functions, Evaluation, and Linear Regression



Legend		Machine Learning A.Y. 2024 - 2025, 2nd Term
Onsite	Major Exam	Instructor: Red M Castilla
Online	No Class	College of Information and Computing Sciences
Project		Schedule: Saturday 8AM - 10AM | 10:30AM - 1:30PM

Date	Day	Week	Session	Activity / Lesson
18-Jan	Saturday	Week #1	Lecture	Overview of Advanced Intelligent Systems
			Laboratory	Pre-Test Examination and Introductions
25-Jan	Saturday	Week #2	Lecture	Loss Functions and Linear Regression
			Laboratory	Summative Lab Exercise #1 Discussion
1-Feb	Saturday	Week #3	Lecture	Linear Classification and Logistic Regression
			Laboratory	Summative Lab Exercise #1 Deadline + Formative Lab Exercise #1
8-Feb	Saturday	Week #4	Lecture	Naïve Bayes and Decision Trees
			Laboratory	Long Examination #1
15-Feb	Saturday	Week #5	Lecture	k-Nearest Neighbors and Support Vector Machines
			Laboratory	Formative Lab Exercise #2

What was discussed in the previous meeting?

What was discussed in the previous meeting?

What is Artificial Intelligence?

What was discussed in the previous meeting?

What is Artificial Intelligence?

Artificial Intelligence is the ***study and creation of machines that perform tasks*** ***normally associated with intelligence****. People from varying backgrounds have their own reasons for interests in AI.*

What was discussed in the previous meeting?

What is Artificial Intelligence?

Artificial Intelligence is the ***study and creation of machines that perform tasks*** ***normally associated with intelligence****. People from varying backgrounds have their own reasons for interests in AI.*

Why is Artificial Intelligence relevant?

What was discussed in the previous meeting?

What is Artificial Intelligence?

Artificial Intelligence is the ***study and creation of machines that perform tasks*** ***normally associated with intelligence****. People from varying backgrounds have their own reasons for interests in AI.*

Why is Artificial Intelligence relevant?

Traditionally human capabilities can be ***undertaken in software inexpensively and at scale****.*
AI can be ***applied to every sector*** *to enable* ***new possibilities and efficiencies****.*

What was discussed in the previous meeting?

What is Artificial Intelligence?

Artificial Intelligence is the ***study and creation of machines that perform tasks*** ***normally associated with intelligence****. People from varying backgrounds have their own reasons for interests in AI.*

Why is Artificial Intelligence relevant?

Traditionally human capabilities can be ***undertaken in software inexpensively and at scale****.*
AI can be ***applied to every sector*** *to enable* ***new possibilities and efficiencies****.*

What is Machine Learning?

What was discussed in the previous meeting?

What is Artificial Intelligence?

Artificial Intelligence is the ***study and creation of machines that perform tasks*** ***normally associated with intelligence****. People from varying backgrounds have their own reasons for interests in AI.*

Why is Artificial Intelligence relevant?

Traditionally human capabilities can be ***undertaken in software inexpensively and at scale****.*
AI can be ***applied to every sector*** *to enable* ***new possibilities and efficiencies****.*

What is Machine Learning?

Machine learning is a ***branch of Artificial Intelligence*** *which focuses on the* ***use of data and algorithms to imitate the way that humans learn****.*

What was discussed in the previous meeting?

What is Artificial Intelligence?

Artificial Intelligence is the ***study and creation of machines that perform tasks*** ***normally associated with intelligence****. People from varying backgrounds have their own reasons for interests in AI.*

Why is Artificial Intelligence relevant?

Traditionally human capabilities can be ***undertaken in software inexpensively and at scale****.*
AI can be ***applied to every sector*** *to enable* ***new possibilities and efficiencies****.*

What is Machine Learning?

Machine learning is a ***branch of Artificial Intelligence*** *which focuses on the* ***use of data and algorithms to imitate the way that humans learn****.*

What goes in the Machine Learning Workflow?

What was discussed in the previous meeting?

What is Artificial Intelligence?

*Artificial Intelligence is the **study and creation of machines that perform tasks** **normally associated with intelligence**. People from varying backgrounds have their own reasons for interests in AI.*

Why is Artificial Intelligence relevant?

*Traditionally human capabilities can be **undertaken in software inexpensively and at scale**. AI can be **applied to every sector** to enable **new possibilities and efficiencies**.*

What is Machine Learning?

*Machine learning is a **branch of Artificial Intelligence** which focuses on the **use of data and algorithms to imitate the way that humans learn**.*

What goes in the Machine Learning Workflow?

***Preprocessing** the data. **Creating models** specific for the task. **Evaluating** if the model has performed with respect to expectations*

Recap on the Machine Learning Setup

What are the elements required for machine learning?

Recap on the Machine Learning Setup

What are the elements required for machine learning?

Output Space

Recap on the Machine Learning Setup

What are the elements required for machine learning?

Output Space

Target variable that is desired to be **estimated**. These are **all possible outputs** that can be generated by the model based on the inputs.

Recap on the Machine Learning Setup

What are the elements required for machine learning?

Output Space

Target variable that is desired to be **estimated**. These are **all possible outputs** that can be generated by the model based on the inputs.

Hypothesis

Recap on the Machine Learning Setup

What are the elements required for machine learning?

Output Space

Target variable that is desired to be **estimated**. These are **all possible outputs** that can be generated by the model based on the inputs.

Hypothesis

Some **speculative relationship** between the input space and the output space. It is **expressed as a collection of parameters characterizing the behavior of the model**.

Recap on the Machine Learning Setup

What are the elements required for machine learning?

Output Space

Target variable that is desired to be **estimated**. These are **all possible outputs** that can be generated by the model based on the inputs.

Hypothesis

Some **speculative relationship** between the input space and the output space. It is **expressed as a collection of parameters characterizing the behavior of the model**.

Input Space

Recap on the Machine Learning Setup

What are the elements required for machine learning?

Output Space

Target variable that is desired to be **estimated**. These are **all possible outputs** that can be generated by the model based on the inputs.

Hypothesis

Some **speculative relationship** between the input space and the output space. It is **expressed as a collection of parameters characterizing the behavior of the model**.

Input Space

This is the **input data**. These can be either called as variables, features, and attributes. The input space comprises **all potential sets of values for input**.

Recap on the Machine Learning Setup

What are the elements required for machine learning?

Output Space

Target variable that is desired to be **estimated**. These are **all possible outputs** that can be generated by the model based on the inputs.

Hypothesis

Some **speculative relationship** between the input space and the output space. It is **expressed as a collection of parameters characterizing the behavior of the model**.

Input Space

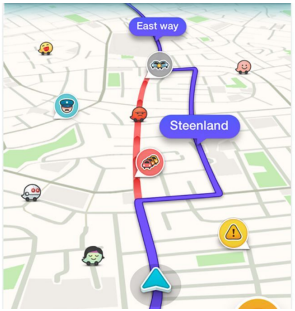
This is the **input data**. These can be either called as variables, features, and attributes. The input space comprises **all potential sets of values for input**.

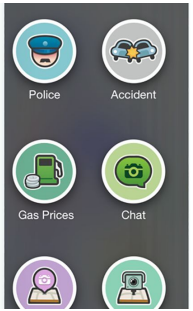
Waze Scenario

Waze Scenario

Scenario:

Suppose you are **travelling** from **your home** to **UST** to attend your morning classes. Your **class is at 9:00 AM** and Waze has **estimated that you will arrive at 8:30AM.** However, due to heavy traffic, **you arrived at 8:50AM**

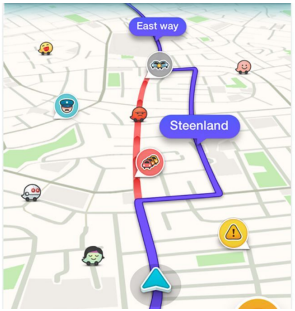


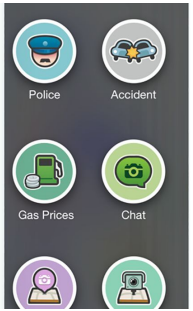


Waze Scenario

Scenario:

Suppose you are **travelling** from **your home** to **UST** to attend your morning classes. Your **class is at 9:00 AM** and Waze has **estimated that you will arrive at 8:30AM.** However, due to heavy traffic, **you arrived at 8:50AM**

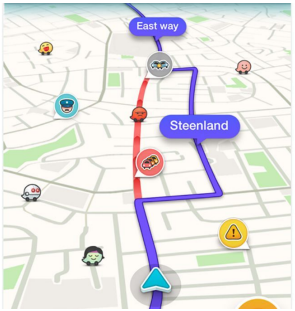


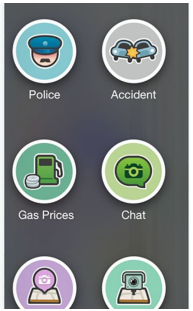


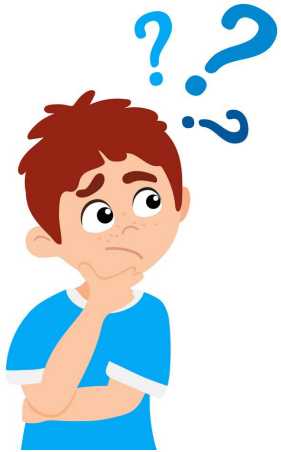
Waze Scenario

Scenario:

Suppose you are **travelling** from **your home** to **UST** to attend your morning classes. Your **class is at 9:00 AM** and Waze has **estimated that you will arrive at 8:30AM.** However, due to heavy traffic, **you arrived at 8:50AM**





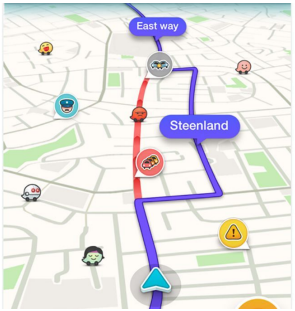


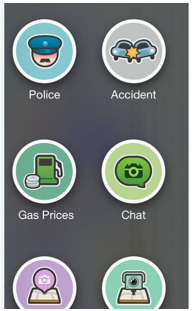
Questions:

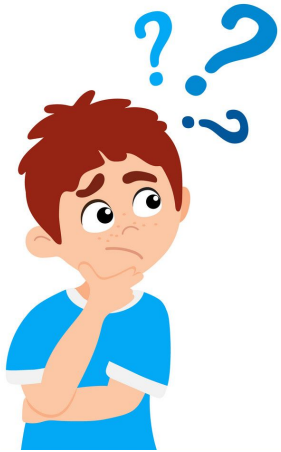
Waze Scenario

Scenario:

Suppose you are **travelling** from **your home** to **UST** to attend your morning classes. Your **class is at 9:00 AM** and Waze has **estimated that you will arrive at 8:30AM.** However, due to heavy traffic, **you arrived at 8:50AM**







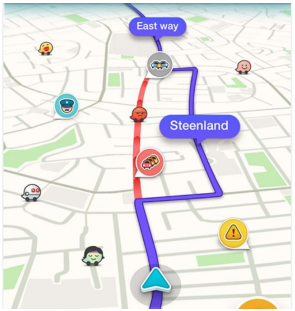
Questions:

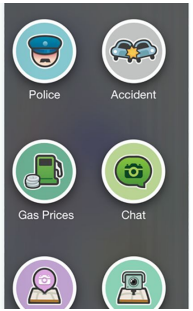
What are were the inputs to Waze?

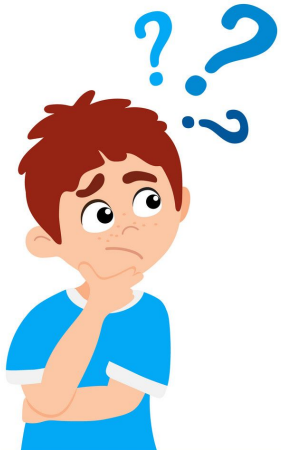
Waze Scenario

Scenario:

Suppose you are **travelling** from **your home** to **UST** to attend your morning classes. Your **class is at 9:00 AM** and Waze has **estimated that you will arrive at 8:30AM.** However, due to heavy traffic, **you arrived at 8:50AM**







Questions:

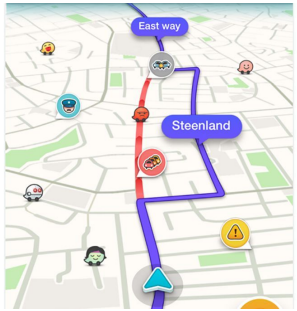
What are were the inputs to Waze?

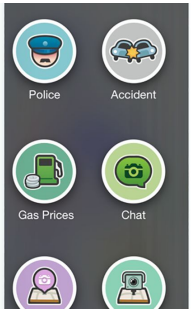
What did Waze estimate?

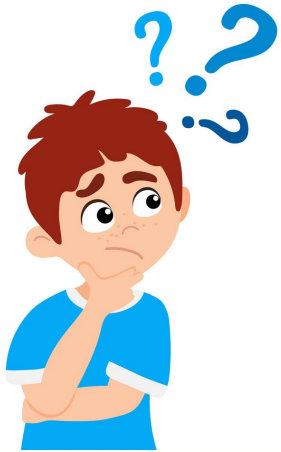
Waze Scenario

Scenario:

Suppose you are **travelling** from **your home** to **UST** to attend your morning classes. Your **class is at 9:00 AM** and Waze has **estimated that you will arrive at 8:30AM.** However, due to heavy traffic, **you arrived at 8:50AM**







Questions:

What are were the inputs to Waze?

What did Waze estimate?

How far was Waze's estimate from the actual time of arrival?

Loss Functions

What are Loss Functions?

Loss Functions

What are Loss Functions?

It is the function that computes the ***distance*** *between the* ***current output*** *of the algorithm and the* ***expected output****.*

Loss Functions

What are Loss Functions?

It is the function that computes the ***distance*** *between the* ***current output*** *of the algorithm and the* ***expected output****.*

More formally…

Loss Functions

What are Loss Functions?

It is the function that computes the ***distance*** *between the* ***current output*** *of the algorithm and the* ***expected output****.*

More formally…

In mathematical optimization and decision theory, a **loss function** or **cost function** (sometimes also called an **error function**) is a function that **maps an event or values of one or more variables onto a real number** intuitively representing some "cost" associated with the event. An **optimization problem seeks to minimize a loss function.**

Loss Functions

Why do we need this?

Loss Functions

Why do we need this?

*It **quantifies the difference** between **predicted** and **actual values** in a machine learning model.*

Loss Functions

Why do we need this?

*It **quantifies the difference** between **predicted** and **actual values** in a machine learning model.*

If you recall from last lecture…

Loss Functions

Why do we need this?

*It **quantifies the difference** between **predicted** and **actual values** in a machine learning model.*

If you recall from last lecture…

Modelling:** Mathematically speaking, a model is **a description of a system using mathematical concepts** and languages. It is a mathematical representation of **objects and their relationships

Loss Functions

Why do we need this?

*It **quantifies the difference** between **predicted** and **actual values** in a machine learning model.*

If you recall from last lecture…

Modelling: *Mathematically speaking, a model is **a description of a system using mathematical concepts** and languages. It is a mathematical representation of **objects and their relationships***

$$Y = f(x)$$

Target Output

Loss Functions

Why do we need this?

*It **quantifies the difference** between **predicted** and **actual values** in a machine learning model.*

If you recall from last lecture…

Modelling: *Mathematically speaking, a model is **a description of a system using mathematical concepts** and languages. It is a mathematical representation of **objects and their relationships***

$$Y = f(x)$$

Target Output **Function**

Loss Functions

Why do we need this?

*It **quantifies the difference** between **predicted** and **actual values** in a machine learning model.*

If you recall from last lecture…

Modelling: *Mathematically speaking, a model is **a description of a system using mathematical concepts** and languages. It is a mathematical representation of **objects and their relationships***

$$Y = f(x)$$

Target Output **Function** **Inputs**

Loss Functions

Why do we need this?

*It **quantifies the difference** between **predicted** and **actual values** in a machine learning model.*

Given the a model output, how to know if it gave the right output?

Loss Functions

Why do we need this?

*It **quantifies the difference** between **predicted** and **actual values** in a machine learning model.*

Given the a model output, how to know if it gave the right output?

Suppose a model predicts if the picture contains a cat or a dog.

Loss Functions

Why do we need this?

*It **quantifies the difference** between **predicted** and **actual values** in a machine learning model.*

Given the a model output, how to know if it gave the right output?

Suppose a model predicts if the picture contains a cat or a dog.

Input

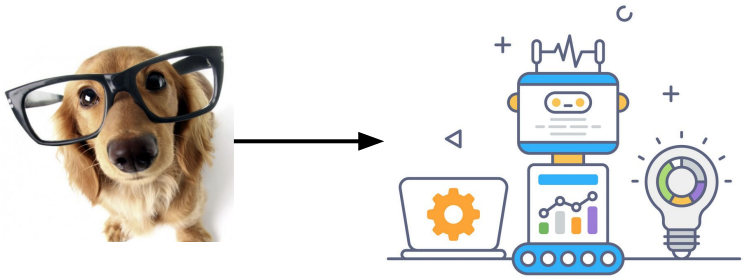
Loss Functions

Why do we need this?

*It **quantifies the difference** between **predicted** and **actual values** in a machine learning model.*

Given the a model output, how to know if it gave the right output?

Suppose a model predicts if the picture contains a cat or a dog.



Input **Model**

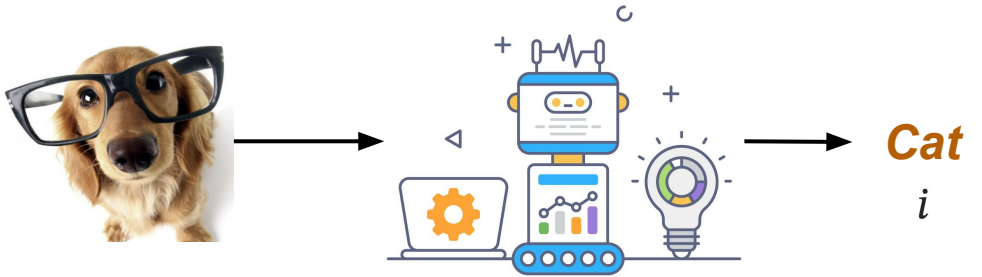
Loss Functions

Why do we need this?

*It **quantifies the difference** between **predicted** and **actual values** in a machine learning model.*

Given the a model output, how to know if it gave the right output?

Suppose a model predicts if the picture contains a cat or a dog.



Input **Model** **Output**

Loss Functions

Why do we need this?

*It **quantifies the difference** between **predicted** and **actual values** in a machine learning model.*

Given the a model output, how to know if it gave the right output?

Suppose a model predicts if the picture contains a cat or a dog.

Input	Model	Output	True Value
		Cat i	Dog j

Loss Functions

Why do we need this?

*It **quantifies the difference** between **predicted** and **actual values** in a machine learning model.*

Given the a model output, how to know if it gave the right output?

Suppose a model predicts if the picture contains a cat or a dog.

Input

Model

Cat

i

Output

Dog

j

True Value

$$L(i,j) = \begin{cases} 0 & i = j \\ 1 & i \neq j \end{cases}$$

Simple Loss Function

Loss Functions

Why do we need this?

*It **quantifies the difference** between **predicted** and **actual values** in a machine learning model.*

Given the a model output, how to know if it gave the right output?

Loss Functions

Why do we need this?

*It **quantifies the difference** between **predicted** and **actual values** in a machine learning model.*

Given the a model output, how to know if it gave the right output?

Suppose a model Waze predicts the estimated time of arrival

Loss Functions

Why do we need this?

*It **quantifies the difference** between **predicted** and **actual values** in a machine learning model.*

Given the a model output, how to know if it gave the right output?

Suppose a model Waze predicts the estimated time of arrival

Input

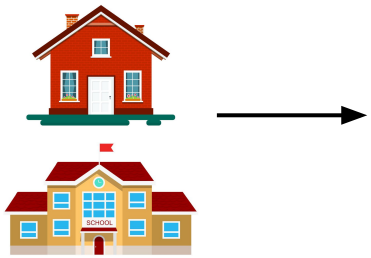
Loss Functions

Why do we need this?

*It **quantifies the difference** between **predicted** and **actual values** in a machine learning model.*

Given the a model output, how to know if it gave the right output?

Suppose a model Waze predicts the estimated time of arrival



Input

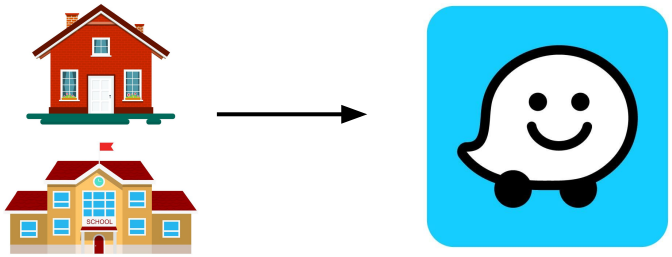
Loss Functions

Why do we need this?

*It **quantifies the difference** between **predicted** and **actual values** in a machine learning model.*

Given the a model output, how to know if it gave the right output?

Suppose a model Waze predicts the estimated time of arrival



Input **Model**

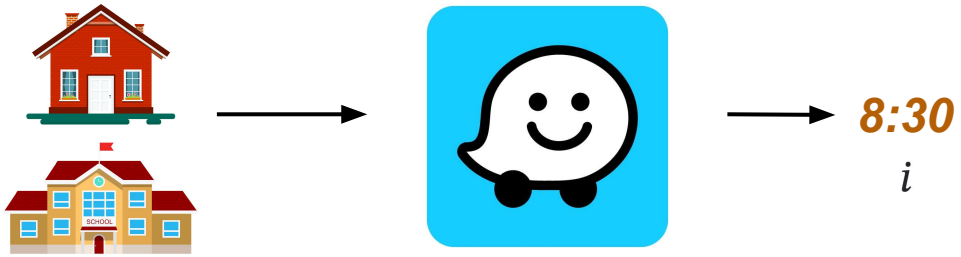
Loss Functions

Why do we need this?

*It **quantifies the difference** between **predicted** and **actual values** in a machine learning model.*

Given the a model output, how to know if it gave the right output?

Suppose a model Waze predicts the estimated time of arrival



Input **Model** **Output**

Loss Functions

Why do we need this?

*It **quantifies the difference** between **predicted** and **actual values** in a machine learning model.*

Given the a model output, how to know if it gave the right output?

Suppose a model Waze predicts the estimated time of arrival

Input

Model

8:30

i

Output

8:50

j

True Value

Loss Functions

Why do we need this?

*It **quantifies the difference** between **predicted** and **actual values** in a machine learning model.*

Given the a model output, how to know if it gave the right output?

Suppose a model Waze predicts the estimated time of arrival

Input

Model

8:30

i

Output

8:50

j

True Value

$$L(i,j) = j - i$$

Simple Loss Function

Loss Functions

Why do we need this?

*It **quantifies the difference** between **predicted** and **actual values** in a machine learning model.*

Loss Functions

Why do we need this?

*It **quantifies the difference** between **predicted** and **actual values** in a machine learning model.*

What are mostly used loss functions?

Regression Problems

Classification Problems

Loss Functions

Why do we need this?

*It **quantifies the difference** between **predicted** and **actual values** in a machine learning model.*

What are mostly used loss functions?

Regression Problems

Classification Problems

Mean Squared Error

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Loss Functions

Why do we need this?

*It **quantifies the difference** between **predicted** and **actual values** in a machine learning model.*

What are mostly used loss functions?

Regression Problems

Classification Problems

Mean Squared Error

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Root Mean Squared Error

$$\mathrm{RMSD} = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \hat{x}_i)^2}{N}}$$

Loss Functions

Why do we need this?

*It **quantifies the difference** between **predicted** and **actual values** in a machine learning model.*

What are mostly used loss functions?

Regression Problems

Classification Problems

Mean Squared Error

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Root Mean Squared Error

$$\text{RMSD} = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \hat{x}_i)^2}{N}}$$

Sum of Squared Error

$$RSS = \sum_{i=1}^{n}(y_i - f(x_i))^2$$

Loss Functions

Why do we need this?

*It **quantifies the difference** between **predicted** and **actual values** in a machine learning model.*

What are mostly used loss functions?

Regression Problems

Mean Squared Error

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Root Mean Squared Error

$$\text{RMSD} = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \hat{x}_i)^2}{N}}$$

Sum of Squared Error

$$RSS = \sum_{i=1}^{n}(y_i - f(x_i))^2$$

Classification Problems

Binary Cross Entropy Loss

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$$

Loss Functions

Why do we need this?

*It **quantifies the difference** between **predicted** and **actual values** in a machine learning model.*

What are mostly used loss functions?

Regression Problems

Mean Squared Error

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Root Mean Squared Error

$$\text{RMSD} = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \hat{x}_i)^2}{N}}$$

Sum of Squared Error

$$RSS = \sum_{i=1}^{n}(y_i - f(x_i))^2$$

Classification Problems

Binary Cross Entropy Loss

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$$

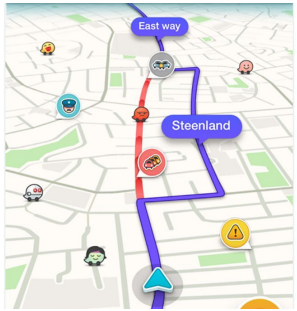
Cross Entropy Loss

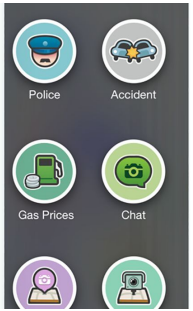
$$L(\hat{y}, y) = -\sum_{k}^{K} y^{(k)} \log \hat{y}^{(k)}$$

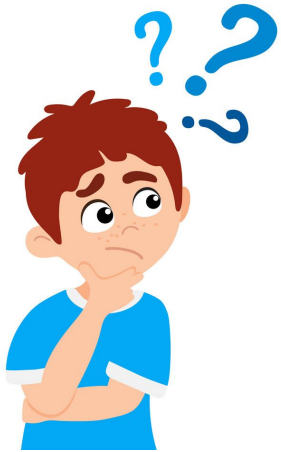
Waze Scenario

Scenario:

Suppose you are **travelling** from **your home** to **UST** to attend your morning classes. Your **class is at 9:00 AM** and Waze has **estimated that you will arrive at 8:30AM.** However, due to heavy traffic, **you arrived at 8:50AM**







Questions:

What are were the inputs to Waze?

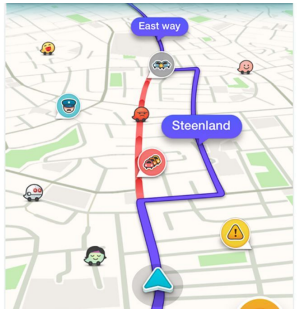
What did Waze estimate?

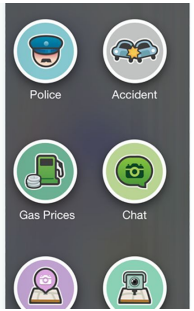
How far was Waze's estimate from the actual time of arrival?

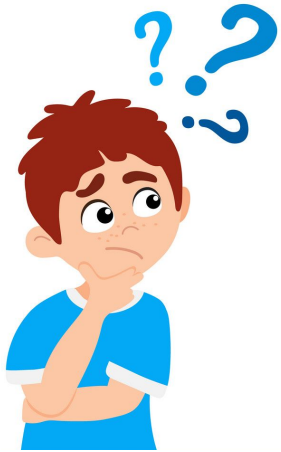
Waze Scenario

Scenario:

Suppose you are **travelling** from **your home** to **UST** to attend your morning classes. Your **class is at 9:00 AM** and Waze has **estimated that you will arrive at 8:30AM.** However, due to heavy traffic, **you arrived at 8:50AM**







Questions:

What are were the inputs to Waze?

What did Waze estimate?

How far was Waze's estimate from the actual time of arrival?

What factors do you think Waze considered for estimating?

Recap on the Machine Learning Setup

What are the elements required for machine learning?

Output Space

Target variable that is desired to be **estimated**. These are **all possible outputs** that can be generated by the model based on the inputs.

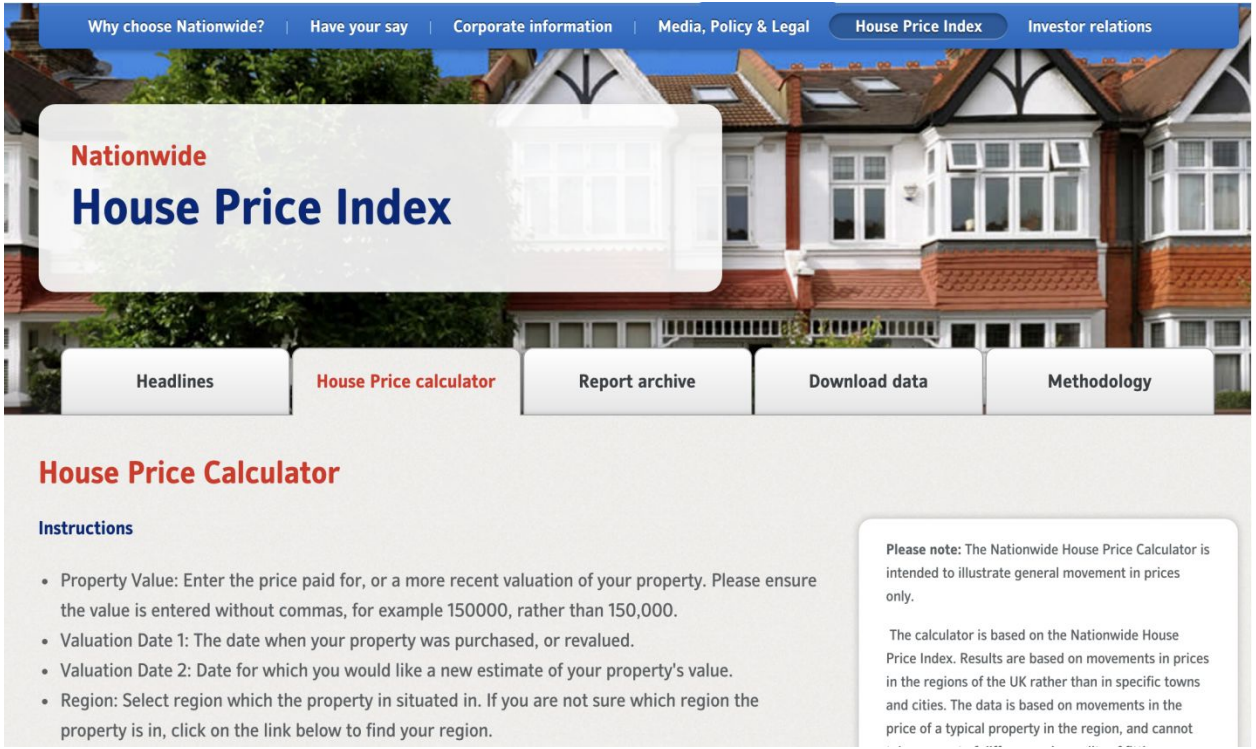
Hypothesis

Some **speculative relationship** between the input space and the output space. It is **expressed as a collection of parameters characterizing the behavior of the model**.

Input Space

This is the **input data**. These can be either called as variables, features, and attributes. The input space comprises **all potential sets of values for input**.

How can we…


Why choose Nationwide? | Have your say | Corporate information | Media, Policy & Legal | House Price Index | Investor relations
Nationwide
House Price Index
Headlines
House Price calculator
Report archive
Download data
Methodology
House Price Calculator
Instructions
Property Value: Enter the price paid for, or a more recent valuation of your property. Please ensure the value is entered without commas, for example 150000, rather than 150,000.
Valuation Date 1: The date when your property was purchased, or revalued.
Valuation Date 2: Date for which you would like a new estimate of your property's value.
Region: Select region which the property in situated in. If you are not sure which region the property is in, click on the link below to find your region.
Please note: The Nationwide House Price Calculator is intended to illustrate general movement in prices only.
The calculator is based on the Nationwide House Price Index. Results are based on movements in prices in the regions of the UK rather than in specific towns and cities. The data is based on movements in the price of a typical property in the region, and cannot

How can we…

What is Regression?

What is Regression?

What is Regression?

What is Regression?

Regression *is a* ***statistical technique*** *that* ***relates a dependent variable*** *to* ***one or more independent*** *variables*

What is Regression?

What is Regression?

Regression *is a* ***statistical technique*** *that* ***relates a dependent variable*** *to* ***one or more independent*** *variables*

What do the problems earlier have in common?

What is Regression?

What is Regression?

***Regression** is a **statistical technique** that **relates a dependent variable** to **one or more independent** variables*

What do the problems earlier have in common?

The targets or predictions are **continuous variables**. (e.g. house prices, stock prices, etc.). We can call them t

What is Regression?

What is Regression?

***Regression** is a **statistical technique** that **relates a dependent variable** to **one or more independent** variables*

What do the problems earlier have in common?

The targets or predictions are **continuous variables**. (e.g. house prices, stock prices, etc.). We can call them t

What do we need to predict these outputs?

What is Regression?

What is Regression?

***Regression** is a **statistical technique** that **relates a dependent variable** to **one or more independent** variables*

What do the problems earlier have in common?

The targets or predictions are **continuous variables**. (e.g. house prices, stock prices, etc.). We can call them t

What do we need to predict these outputs?

Features: These are the inputs. We can call them x (or $\mathbf{X}$ if vector)

What is Regression?

What is Regression?

***Regression** is a **statistical technique** that **relates a dependent variable** to **one or more independent** variables*

What do the problems earlier have in common?

The targets or predictions are **continuous variables**. (e.g. house prices, stock prices, etc.). We can call them t

What do we need to predict these outputs?

Features: These are the inputs. We can call them x (or $\mathbf{X}$ if vector)

Training Samples: Many samples of $x^{(i)}$ for which $t^{(i)}$ is known

What is Regression?

What is Regression?

***Regression** is a **statistical technique** that **relates a dependent variable** to **one or more independent** variables*

What do the problems earlier have in common?

The targets or predictions are **continuous variables**. (e.g. house prices, stock prices, etc.). We can call them t

What do we need to predict these outputs?

Features: These are the inputs. We can call them x (or $\mathbf{x}$ if vector)

Training Samples: Many samples of $x^{(i)}$ for which $t^{(i)}$ is known

Model: Function that models relationship between x and t

What is Regression?

What is Regression?

***Regression** is a **statistical technique** that **relates a dependent variable** to **one or more independent** variables*

What do the problems earlier have in common?

The targets or predictions are **continuous variables**. (e.g. house prices, stock prices, etc.). We can call them t

What do we need to predict these outputs?

Features: These are the inputs. We can call them x (or $\mathbf{X}$ if vector)

Training Samples: Many samples of $x^{(i)}$ for which $t^{(i)}$ is known

Model: Function that models relationship between x and t

Loss: Tells how well the model approximates the target, given the training examples

What is Regression?

What is Regression?

***Regression** is a **statistical technique** that **relates a dependent variable** to **one or more independent** variables*

What do the problems earlier have in common?

The targets or predictions are **continuous variables**. (e.g. house prices, stock prices, etc.). We can call them t

What do we need to predict these outputs?

Features: These are the inputs. We can call them x (or $\mathbf{X}$ if vector)

Training Samples: Many samples of $x^{(i)}$ for which $t^{(i)}$ is known

Model: Function that models relationship between x and t

Loss: Tells how well the model approximates the target, given the training examples

Optimization: A way of finding the parameters of our model that minimizes the loss function

What is Regression?

What is Regression?

Regression *is a* ***statistical technique*** *that* ***relates a dependent variable*** *to* ***one or more independent*** *variables*

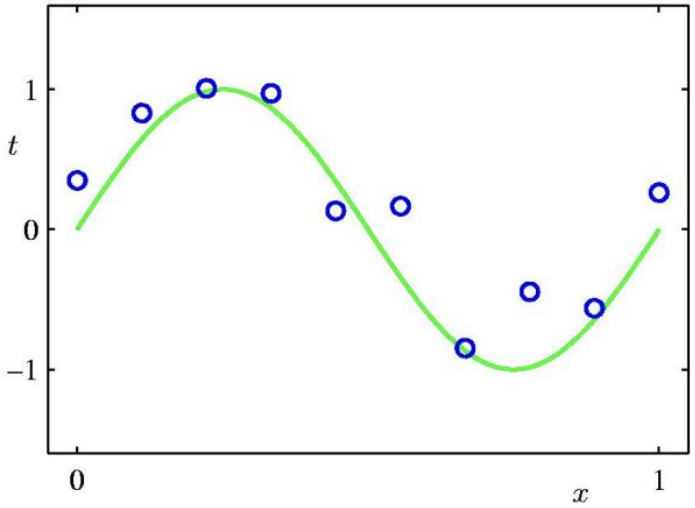
Here's a Simple 1-D Regression…

What is Regression?

What is Regression?

Regression *is a* ***statistical technique*** *that* ***relates a dependent variable*** *to* ***one or more independent*** *variables*

Here's a Simple 1-D Regression…



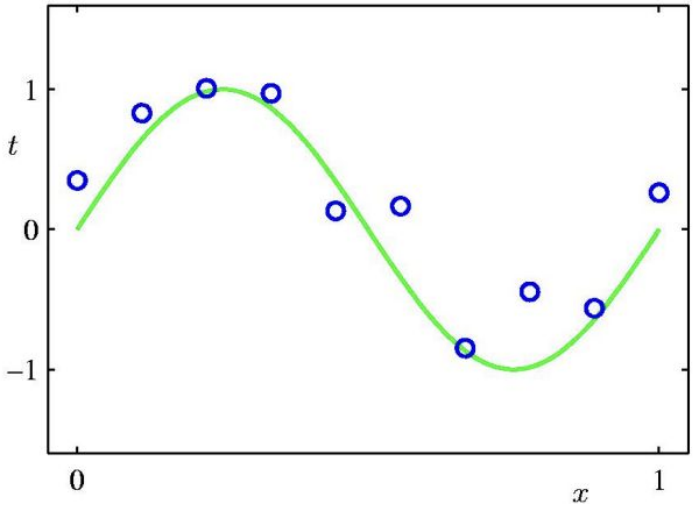
What is Regression?

What is Regression?

Regression *is a* ***statistical technique*** *that* ***relates a dependent variable*** *to* ***one or more independent*** *variables*

Here's a Simple 1-D Regression…

Circles are **data points** (i.e. training examples) that are provided



What is Regression?

What is Regression?

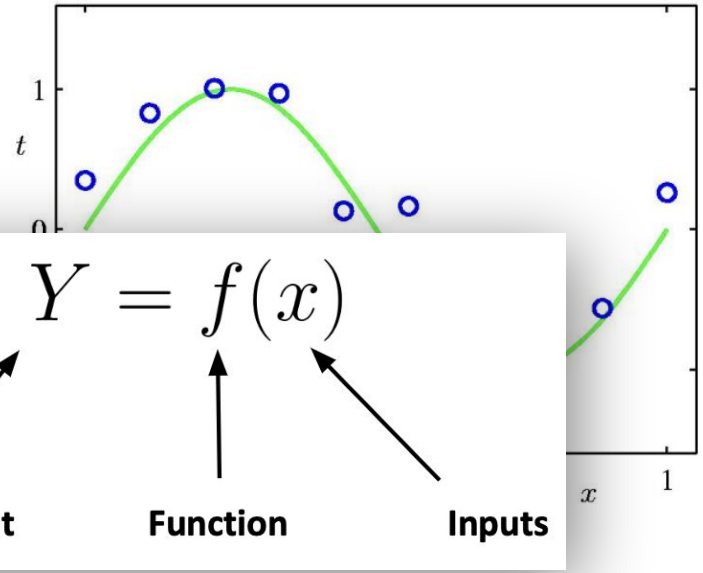
***Regression** is a **statistical technique** that **relates a dependent variable** to **one or more independent** variables*

Here's a Simple 1-D Regression…

Circles are **data points** (i.e. training examples) that are provided

The data points are uniform in x but may be displaced in y with some noise ϵ

$$t(x) = f(x) + \epsilon$$



Recall $Y = f(x)$

Target Output **Function** **Inputs**

What is Regression?

What is Regression?

***Regression** is a **statistical technique** that **relates a dependent variable** to **one or more independent** variables*

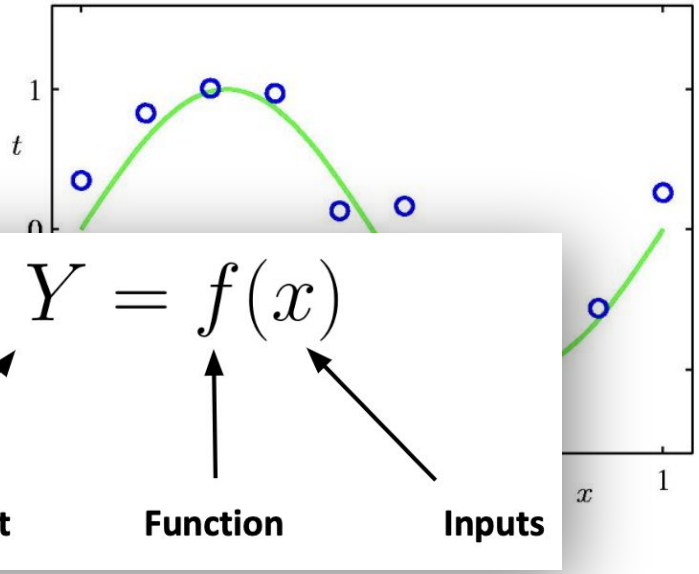
Here's a Simple 1-D Regression…

Circles are **data points** (i.e. training examples) that are provided

The data points are uniform in x but may be displaced in y with some noise ϵ

$$t(x) = f(x) + \epsilon$$

The function *f* is the model that the algorithm wants to estimate.



Recall

$$Y = f(x)$$

Target Output **Function** **Inputs**

What is Regression?

What is Regression?

***Regression** is a **statistical technique** that **relates a dependent variable** to **one or more independent** variables*

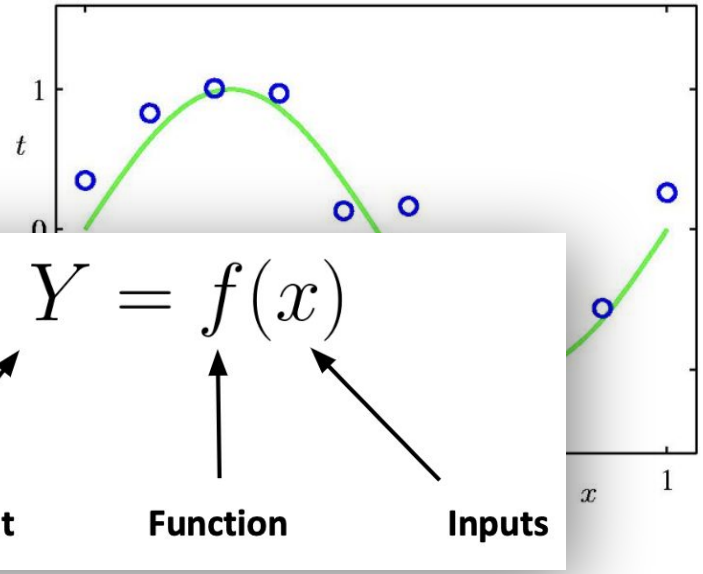
Here's a Simple 1-D Regression…

Circles are **data points** (i.e. training examples) that are provided

The data points are uniform in x but may be displaced in y with some noise ϵ

$$t(x) = f(x) + \epsilon$$

There is a certain error or noise term here that doesn't allow us to perfectly estimate the target outputs



Recall $Y = f(x)$

Target Output **Function** **Inputs**

What is Regression?

What is Regression?

***Regression** is a **statistical technique** that **relates a dependent variable** to **one or more independent** variables*

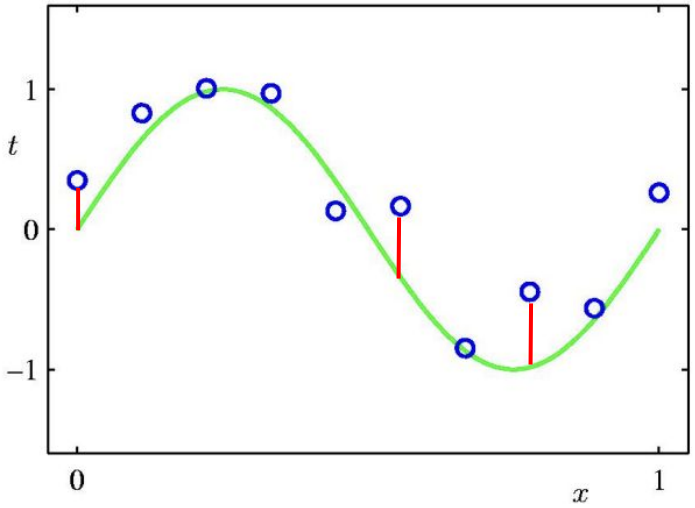
Here's a Simple 1-D Regression…

Circles are **data points** (i.e. training examples) that are provided

The data points are uniform in x but may be displaced in y with some noise ϵ

$$t(x) = f(x) + \epsilon$$

There is a certain error or noise term here that doesn't allow us to perfectly estimate the target outputs



What is Regression?

What is Regression?

***Regression** is a **statistical technique** that **relates a dependent variable** to **one or more independent** variables*

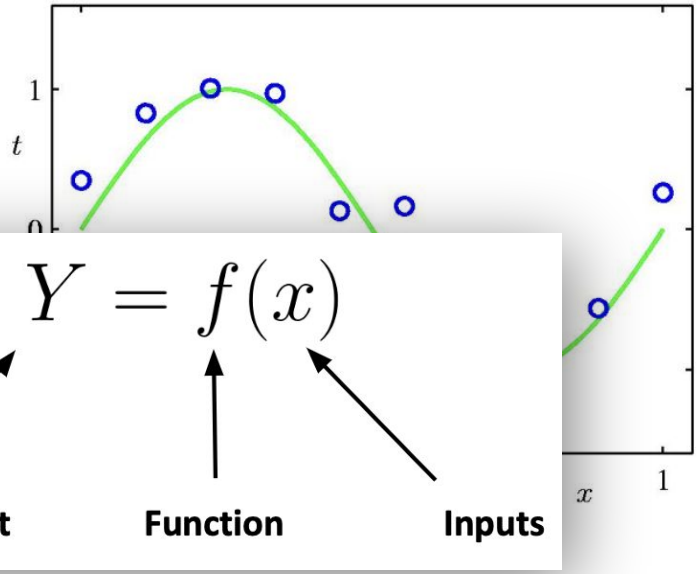
Here's a Simple 1-D Regression…

Circles are **data points** (i.e. training examples) that are provided

The data points are uniform in x but may be displaced in y with some noise ϵ

$$t(x) = f(x) + \epsilon$$

The target given that there is also error involved since the algorithm isn't perfect



Recall $Y = f(x)$

Target Output **Function** **Inputs**

What is Regression?

What is Regression?

***Regression** is a **statistical technique** that **relates a dependent variable** to **one or more independent** variables*

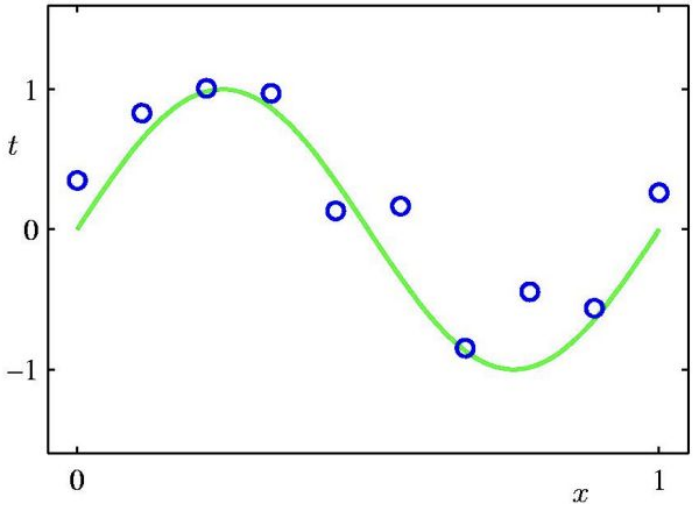
Here's a Simple 1-D Regression…

Circles are **data points** (i.e. training examples) that are provided

The data points are uniform in x but may be displaced in y with some noise ϵ

$$t(x) = f(x) + \epsilon$$

The **green** is the **true curve** that **we don't know**



What is Regression?

What is Regression?

***Regression** is a **statistical technique** that **relates a dependent variable** to **one or more independent** variables*

Here's a Simple 1-D Regression…

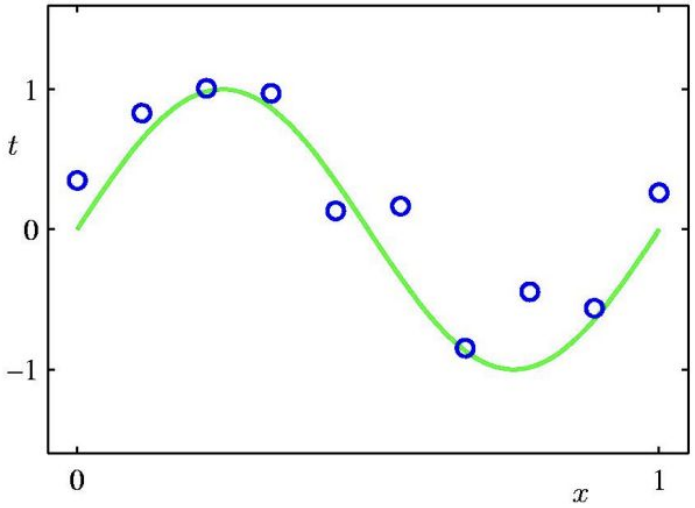
Circles are **data points** (i.e. training examples) that are provided

The data points are uniform in x but may be displaced in y with some noise ϵ

$$t(x) = f(x) + \epsilon$$

The **green** is the **true curve** that **we don't know**

Goal: We want to **fit a curve** to the **data points**



Working Example: Boston Housing Data

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Working Example: Boston Housing Data

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

- ***Features***
- ***Training Data***
- ***Model***
- ***Loss***
- ***Optimization***

Working Example: Boston Housing Data

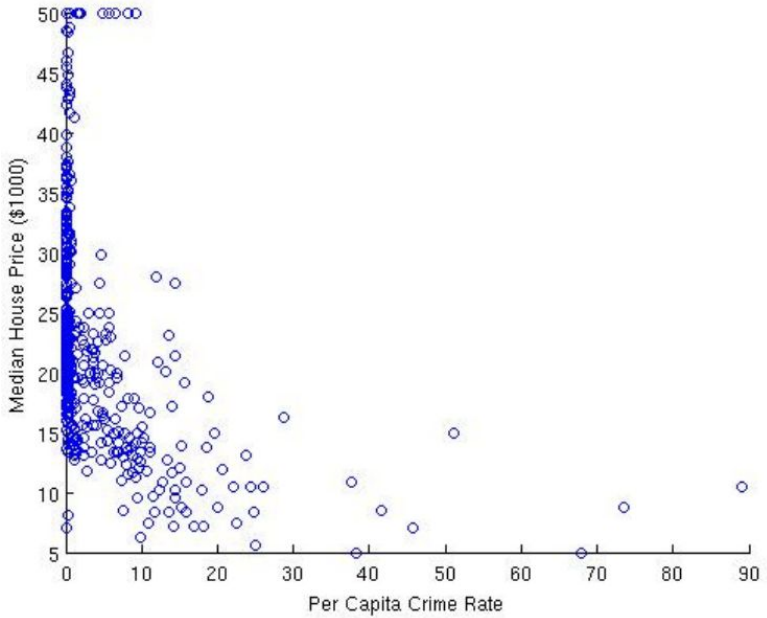
Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

- ***Features***
- *Training Data*
- *Model*
- *Loss*
- *Optimization*

Working Example: Boston Housing Data

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

- ***Features***
- *Training Data*
- *Model*
- *Loss*
- *Optimization*



First possible feature: Per Capita Crime Rate

Working Example: Boston Housing Data

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

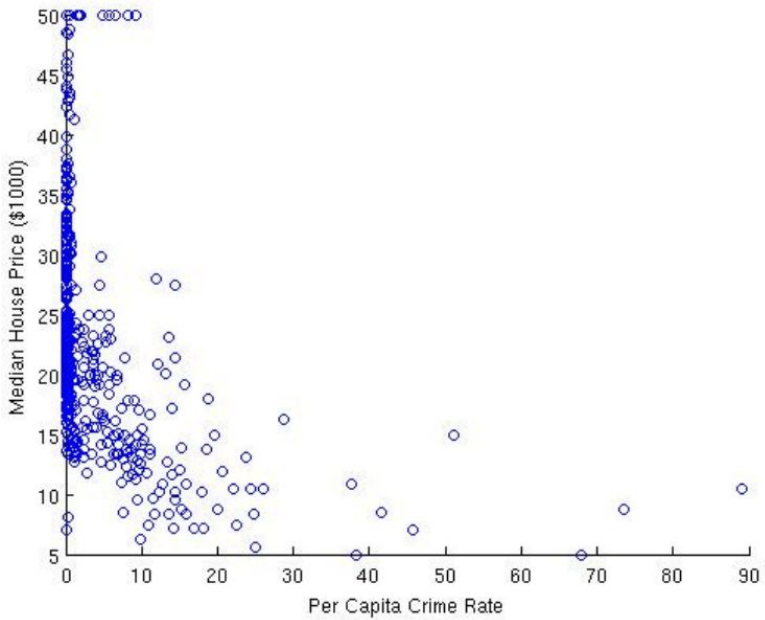
Features

Training Data

Model

Loss

Optimization



First possible feature: Per Capita Crime Rate

Do you think this is a **good feature?**

Working Example: Boston Housing Data

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

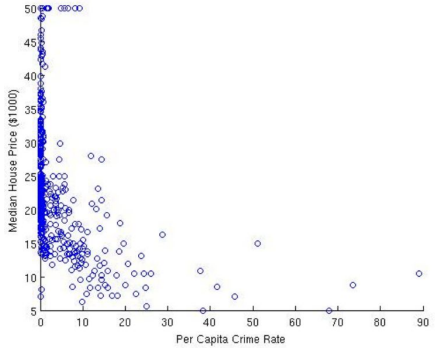
Training Data

Model

Loss

Optimization

How to represent the data?



Working Example: Boston Housing Data

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

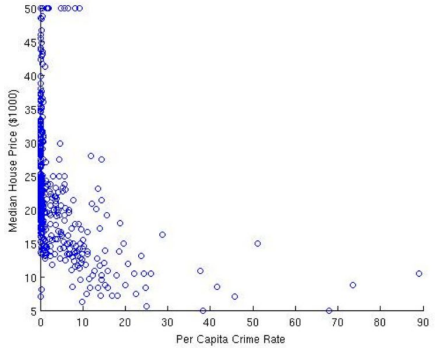
Model

Loss

Optimization

How to represent the data?

Data is described as pairs $D = \{(x^{(1)}, t^{(1)}), \cdots, (x^{(N)}, t^{(N)})\}$



Working Example: Boston Housing Data

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

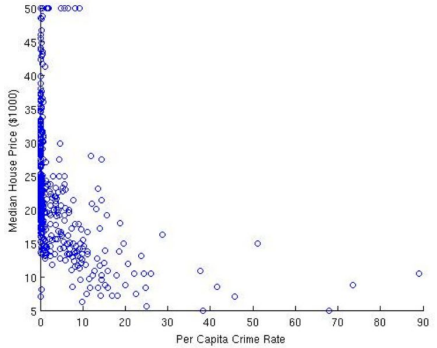
Loss

Optimization

How to represent the data?

Data is described as pairs $D = \{(x^{(1)}, t^{(1)}), \cdots, (x^{(N)}, t^{(N)})\}$

$x \in \mathbb{R}$ is the **input feature** (per capita crime rate)



Working Example: Boston Housing Data

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

Optimization

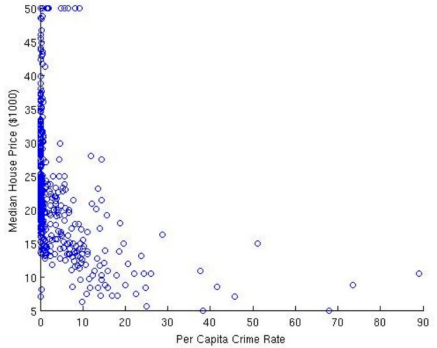
How to represent the data?

Data is described as pairs $D = \{(x^{(1)}, t^{(1)}), \cdots, (x^{(N)}, t^{(N)})\}$

$x \in \mathbb{R}$ is the **input feature** (per capita crime rate)

$t \in \mathbb{R}$ is the **target output** (median house price)

(i) simply **indicates the training examples** (we have *N* examples)



How does it look like for our example?

The **Median House Price** is t

The **Per Capita Crime Rate** is x

Each **dot in the plot is one data point** $(x^{(1)}, t^{(1)}), \cdots$

Working Example: Boston Housing Data

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

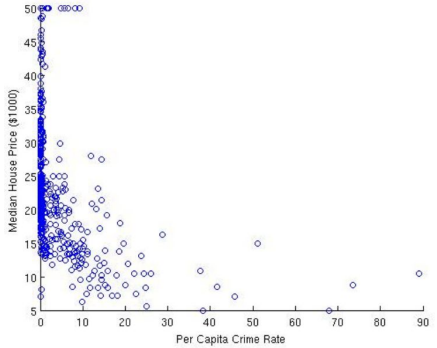
Optimization

How to represent the data?

Data is described as pairs $D = \{(x^{(1)}, t^{(1)}), \cdots, (x^{(N)}, t^{(N)})\}$

$x \in \mathbb{R}$ is the **input feature** (per capita crime rate)

$t \in \mathbb{R}$ is the **target output** (median house price)



Working Example: Boston Housing Data

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

Optimization

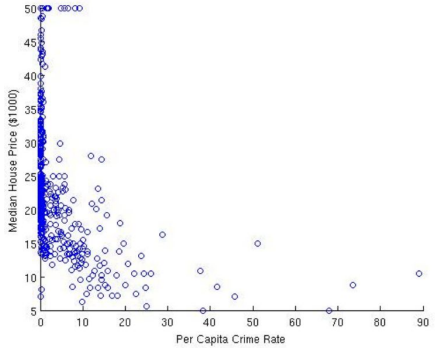
How to represent the data?

Data is described as pairs $D = \{(x^{(1)}, t^{(1)}), \cdots, (x^{(N)}, t^{(N)})\}$

$x \in \mathbb{R}$ is the **input feature** (per capita crime rate)

$t \in \mathbb{R}$ is the **target output** (median house price)

(i) simply **indicates the training examples** (we have *N* examples)



Working Example: Boston Housing Data

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

Optimization

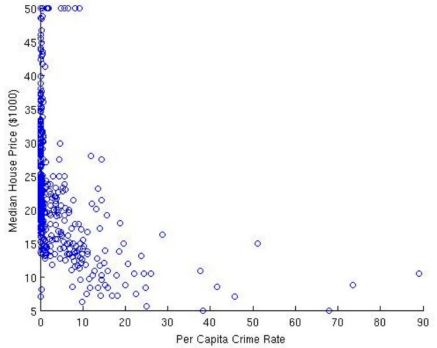
How to represent the data?

Data is described as pairs $D = \{(x^{(1)}, t^{(1)}), \cdots, (x^{(N)}, t^{(N)})\}$

$x \in \mathbb{R}$ is the **input feature** (per capita crime rate)

$t \in \mathbb{R}$ is the **target output** (median house price)

(i) simply **indicates the training examples** (we have *N* examples)



How does it look like for our example?

Working Example: Boston Housing Data

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

Optimization

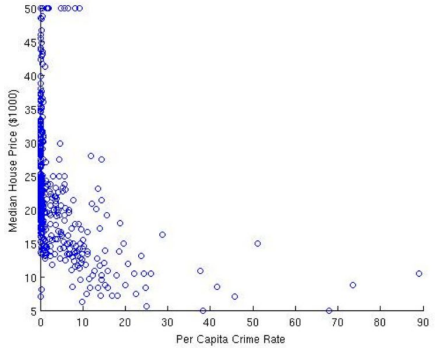
How to represent the data?

Data is described as pairs $D = \{(x^{(1)}, t^{(1)}), \cdots, (x^{(N)}, t^{(N)})\}$

$x \in \mathbb{R}$ is the **input feature** (per capita crime rate)

$t \in \mathbb{R}$ is the **target output** (median house price)

(i) simply **indicates the training examples** (we have *N* examples)



How does it look like for our example?

The **Median House Price** is t

Working Example: Boston Housing Data

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

Optimization

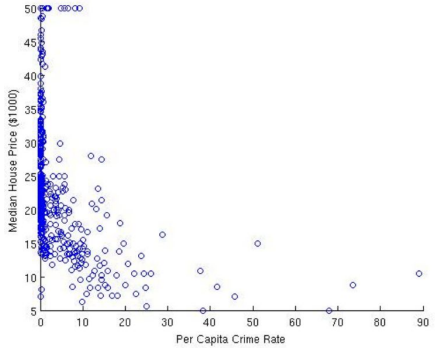
How to represent the data?

Data is described as pairs $D = \{(x^{(1)}, t^{(1)}), \cdots, (x^{(N)}, t^{(N)})\}$

$x \in \mathbb{R}$ is the **input feature** (per capita crime rate)

$t \in \mathbb{R}$ is the **target output** (median house price)

(i) simply **indicates the training examples** (we have *N* examples)



How does it look like for our example?

The **Median House Price** is t

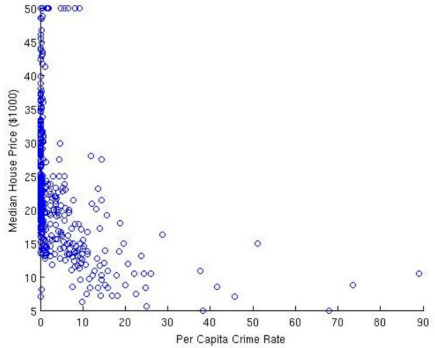
The **Per Capita Crime Rate** is x

Working Example: Boston Housing Data

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

- *Features*
- *Training Data*
- ***Model***
- *Loss*
- *Optimization*

What model do we use?



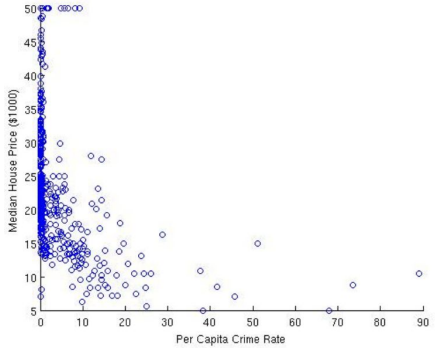
Working Example: Boston Housing Data

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

- *Features*
- *Training Data*
- ***Model***
- *Loss*
- *Optimization*

What model do we use?

We have 1 feature / variable. Is there any model that you can think of?



Working Example: Boston Housing Data

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

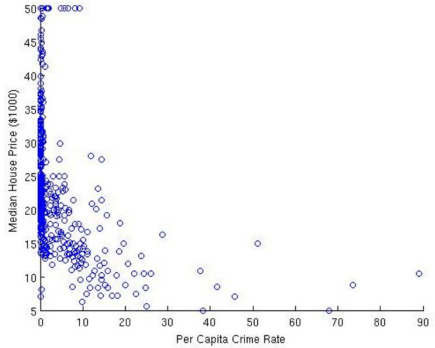
Loss

Optimization

What model do we use?

We have 1 feature / variable. Is there any model that you can think of?

$$y(x) = w_0 + w_1 x$$



Working Example: Boston Housing Data

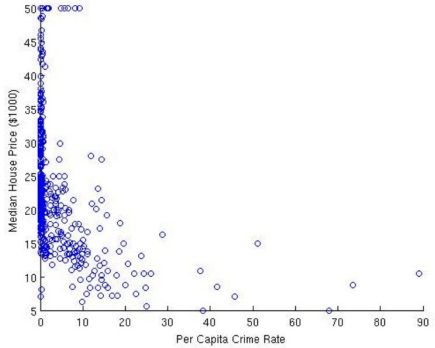
Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

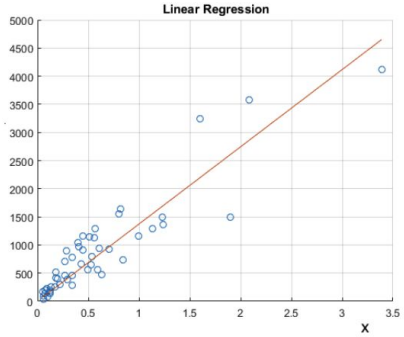
- *Features*
- *Training Data*
- ***Model***
- *Loss*
- *Optimization*

What model do we use?

We have 1 feature / variable. Is there any model that you can think of?

$$y(x) = w_0 + w_1 x$$





Working Example: Boston Housing Data

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

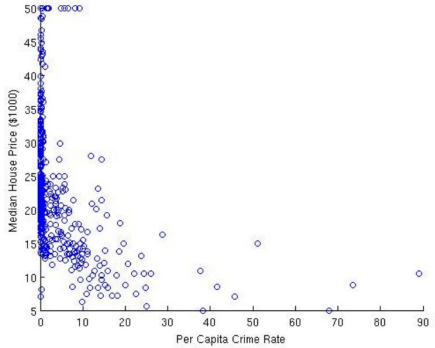
Loss

Optimization

What model do we use?

We have 1 feature / variable. Is there any model that you can think of?

$$y(x) = w_0 + w_1 x$$



Working Example: Boston Housing Data

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

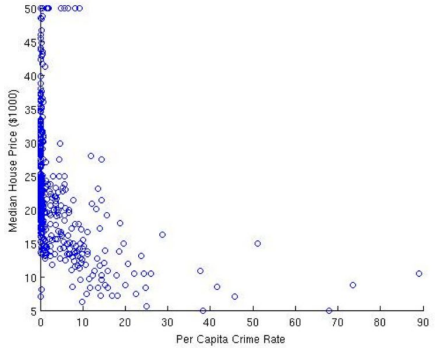
Loss

Optimization

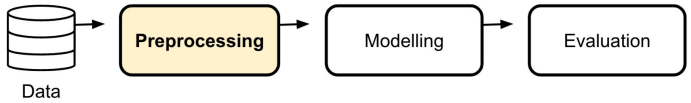
What model do we use?

We have 1 feature / variable. Is there any model that you can think of?

$$y(x) = w_0 + w_1 x$$



The Usual Machine Learning Workflow



Working Example: Boston Housing Data

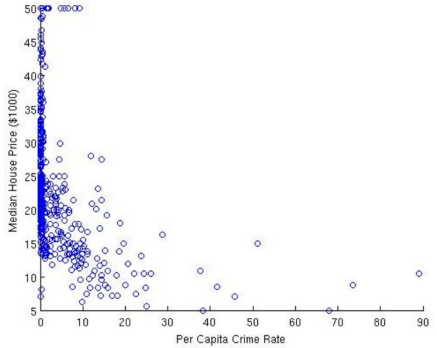
Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

- *Features*
- *Training Data*
- ***Model***
- *Loss*
- *Optimization*

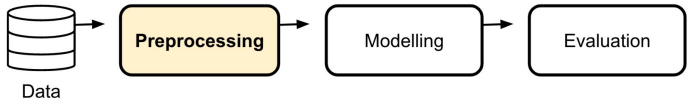
What model do we use?

We have 1 feature / variable. Is there any model that you can think of?

$$y(x) = w_0 + w_1 x$$



The Usual Machine Learning Workflow



In practice, at this point, the data should have been split already to **training and testing sets**. The model should map the input x to y

Working Example: Boston Housing Data

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

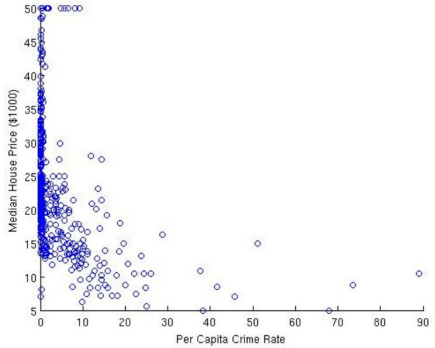
Training Data

Model

Loss

Optimization

What about noise?



Working Example: Boston Housing Data

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

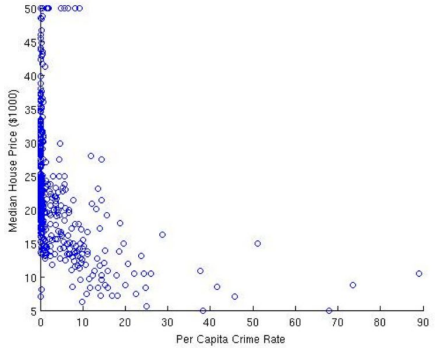
Model

Loss

Optimization

What about noise?

A simple model typically **does not exactly fit the data** – lack of fit can be considered **noise**



Working Example: Boston Housing Data

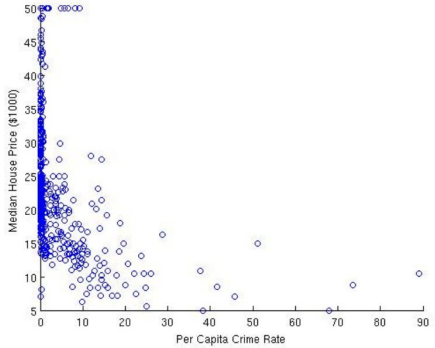
Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

- *Features*
- *Training Data*
- ***Model***
- *Loss*
- *Optimization*

What about noise?

A simple model typically **does not exactly fit the data** – lack of fit can be considered **noise**

What are the sources of noise?



Working Example: Boston Housing Data

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

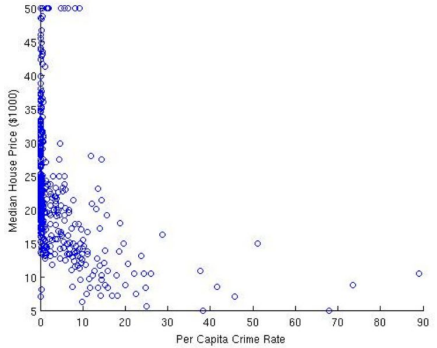
Optimization

What about noise?

A simple model typically **does not exactly fit the data** – lack of fit can be considered **noise**

What are the sources of noise?

Imprecision in data Attributes (input noise)



Working Example: Boston Housing Data

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

- *Features*
- *Training Data*
- ***Model***
- *Loss*
- *Optimization*

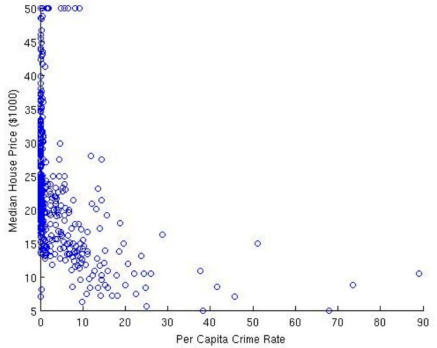
What about noise?

A simple model typically **does not exactly fit the data** – lack of fit can be considered **noise**

What are the sources of noise?

Imprecision in data Attributes (input noise)

Errors in data targets (mislabelling)



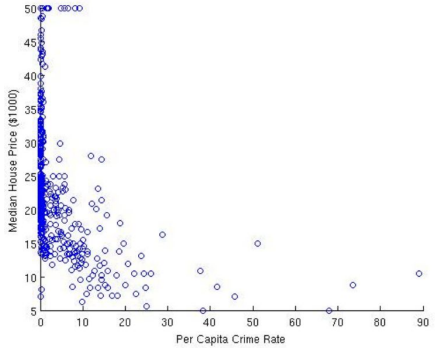
Working Example: Boston Housing Data

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

- *Features*
- *Training Data*
- ***Model***
- *Loss*
- *Optimization*

What about noise?

A simple model typically **does not exactly fit the data** – lack of fit can be considered **noise**



What are the sources of noise?

Imprecision in data Attributes (input noise)

Errors in data targets (mislabelling)

Additional attributes **not taken into account** by data attributes, affect target values (latent variables)

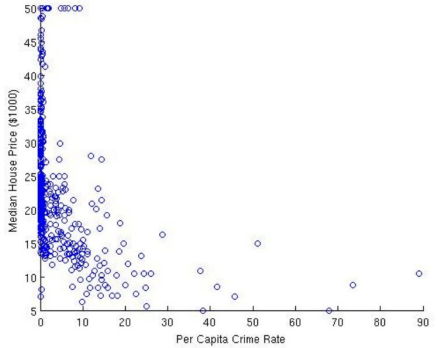
Working Example: Boston Housing Data

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

- *Features*
- *Training Data*
- ***Model***
- *Loss*
- *Optimization*

What about noise?

A simple model typically **does not exactly fit the data** – lack of fit can be considered **noise**



What are the sources of noise?

Imprecision in data Attributes (input noise)

Errors in data targets (mislabelling)

Additional attributes **not taken into account** by data attributes, affect target values (latent variables)

Model may be too simple to account for data targets

Identifying the Loss Function

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

What loss do we use?

Features

Training Data

Model

Loss

Optimization

Identifying the Loss Function

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

Optimization

What loss do we use?

$$y(x) = w_0 + w_1 x$$

Identifying the Loss Function

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

Optimization

What loss do we use?

$$y(x) = w_0 + w_1 x$$

Identifying the Loss Function

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

Optimization

What loss do we use?

$$y(x) = w_0 + w_1 x$$

Using Sum of Squared Error Loss

Identifying the Loss Function

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

Optimization

What loss do we use?

$$y(x) = w_0 + w_1 x$$

Using Sum of Squared Error Loss

The loss function measures the **squared error between true labels**

Identifying the Loss Function

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

Optimization

What loss do we use?

$$y(x) = w_0 + w_1 x$$

Using Sum of Squared Error Loss

The loss function measures the **squared error between true labels**

$$\ell(\mathbf{w}) = \sum_{n=1}^{N} [t^{(n)} - (w_0 + w_1 x^{(n)})]^2$$

Identifying the Loss Function

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

Optimization

What loss do we use?

$$y(x) = w_0 + w_1 x$$

Using Sum of Squared Error Loss

The loss function measures the **squared error between true labels**

$$\ell(\mathbf{w}) = \sum_{n=1}^{N} [t^{(n)} - (w_0 + w_1 x^{(n)})]^2$$

This is the model that predicts the output given feature *x*

Identifying the Loss Function

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

Optimization

What loss do we use?

$$y(x) = w_0 + w_1 x$$

Using Sum of Squared Error Loss

The loss function measures the **squared error between true labels**

$$\ell(\mathbf{w}) = \sum_{n=1}^{N} [t^{(n)} - (w_0 + w_1 x^{(n)})]^2$$

This is the true labels of the example

Identifying the Loss Function

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

Optimization

What loss do we use?

$$y(x) = w_0 + w_1 x$$

Using Sum of Squared Error Loss

The loss function measures the **squared error between true labels**

$$\ell(\mathbf{w}) = \sum_{n=1}^{N} [t^{(n)} - (w_0 + w_1 x^{(n)})]^2$$

Why?

Identifying the Loss Function

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

Optimization

What loss do we use?

$$y(x) = w_0 + w_1 x$$

Using Sum of Squared Error Loss

The loss function measures the **squared error between true labels**

$$\ell(\mathbf{w}) = \sum_{n=1}^{N} [t^{(n)} - (w_0 + w_1 x^{(n)})]^2$$

Why do we need to square the difference?

Identifying the Loss Function

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

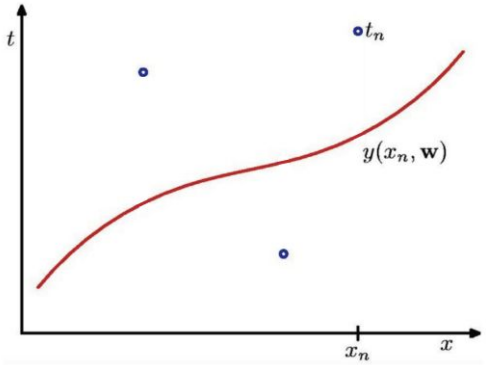
Model

Loss

Optimization

What loss do we use?

$$y(x) = w_0 + w_1 x$$



Using Sum of Squared Error Loss

The loss function measures the **squared error between true labels**

$$\ell(\mathbf{w}) = \sum_{n=1}^{N} [t^{(n)} - (w_0 + w_1 x^{(n)})]^2$$

For a particular hypothesis (y(x) defined by a choice of ***w***, drawn in red), **what does the loss represent geometrically**?

Identifying the Loss Function

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

Optimization

What loss do we use?

$$y(x) = w_0 + w_1 x$$

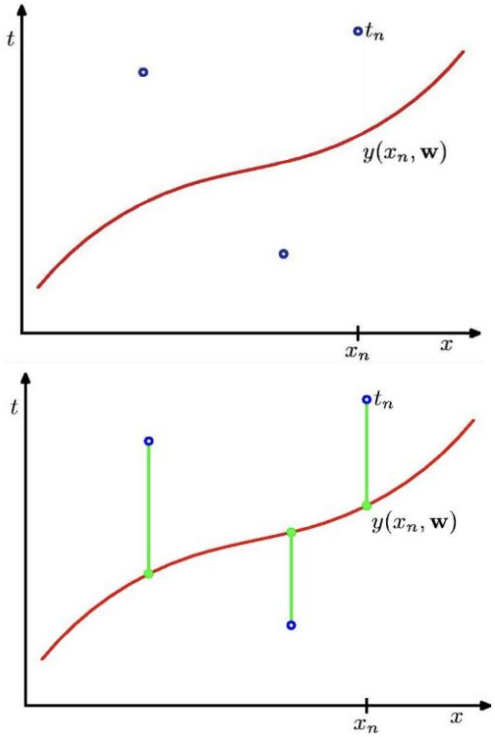
Using Sum of Squared Error Loss

The loss function measures the **squared error between true labels**

$$\ell(\mathbf{w}) = \sum_{n=1}^{N} [t^{(n)} - (w_0 + w_1 x^{(n)})]^2$$

For a particular hypothesis (y(x) defined by a choice of ***w***, drawn in red), **what does the loss represent geometrically**?

The **loss for the red hypothesis is the sum of the squared vertical errors**



Optimizing the Objective

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

How to figure out the weights?

Features

Training Data

Model

Loss

Optimization

Optimizing the Objective

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

Optimization

How to figure out the weights?

$$y(x) = w_0 + w_1 x$$

Optimizing the Objective

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

Optimization

How to figure out the weights?

$$y(x) = w_0 + w_1 x$$

We **need to find weights** $\mathbf{w}$ such that it minimizes the loss $\ell(\mathbf{w})$

Optimizing the Objective

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

Optimization

How to figure out the weights?

$$y(x) = w_0 + w_1 x$$

We **need to find weights** $\mathbf{w}$ such that it minimizes the loss $\ell(\mathbf{w})$

Using Gradient Descent for one example

Optimizing the Objective

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

Optimization

How to figure out the weights?

$$y(x) = w_0 + w_1 x$$

We **need to find weights** $\mathbf{w}$ such that it minimizes the loss $\ell(\mathbf{w})$

Using Gradient Descent for one example

Initialize $\mathbf{w}$ (e.g. random initialization)

Sample: $y(x) = 0.5 + 0.25x$

Optimizing the Objective

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

Optimization

How to figure out the weights?

$$y(x) = w_0 + w_1 x$$

We **need to find weights** $\mathbf{w}$ such that it minimizes the loss $\ell(\mathbf{w})$

Using Gradient Descent for one example

Initialize $\mathbf{w}$ (e.g. random initialization)

Repeatedly update $\mathbf{w}$ based on the gradient

Sample: $y(x) = 0.5 + 0.25x$

Sample Update: $y(x) = 0.60 + 0.75x$

Optimizing the Objective

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

Optimization

How to figure out the weights?

$$y(x) = w_0 + w_1 x$$

We **need to find weights** $\mathbf{w}$ such that it minimizes the loss $\ell(\mathbf{w})$

Using Gradient Descent for one example

Initialize $\mathbf{w}$ (e.g. random initialization)

Repeatedly update $\mathbf{w}$ based on the gradient

$$\mathbf{w} \leftarrow \mathbf{w} - \lambda \frac{\partial \ell}{\partial \mathbf{w}}$$

λ is the **learning rate**

Sample: $y(x) = 0.5 + 0.25x$

Sample Update: $y(x) = 0.60 + 0.75x$

Optimizing the Objective

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

Optimization

How to figure out the weights?

$$y(x) = w_0 + w_1 x$$

We **need to find weights** $\mathbf{w}$ such that it minimizes the loss $\ell(\mathbf{w})$

Using Gradient Descent for one example

Initialize $\mathbf{w}$ (e.g. random initialization)

Repeatedly update $\mathbf{w}$ based on the gradient

$$\mathbf{w} \leftarrow \mathbf{w} - \lambda \frac{\partial \ell}{\partial \mathbf{w}}$$

λ is the **learning rate**

Partial derivative of the loss

Sample: $y(x) = 0.5 + 0.25x$

Sample Update: $y(x) = 0.60 + 0.75x$

Optimizing the Objective

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

Optimization

How to figure out the weights?

$$y(x) = w_0 + w_1 x$$

We **need to find weights** $\mathbf{w}$ such that it minimizes the loss $\ell(\mathbf{w})$

Using Gradient Descent for one example

Initialize $\mathbf{w}$ (e.g. random initialization)

Sample: $y(x) = 0.5 + 0.25x$

Sample Update: $y(x) = 0.60 + 0.75x$

Repeatedly update $\mathbf{w}$ based on the gradient

$$\mathbf{w} \leftarrow \mathbf{w} - \lambda \frac{\partial \ell}{\partial \mathbf{w}}$$

λ is the **learning rate**

Partial derivative of the loss ***with respect to weights***

Optimizing the Objective

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

- *Features*
- *Training Data*
- *Model*
- *Loss*
- ***Optimization***

How to figure out the weights?

$$y(x) = w_0 + w_1 x$$

We **need to find weights** $\mathbf{w}$ such that it minimizes the loss $\ell(\mathbf{w})$

Using Gradient Descent for one example

Initialize $\mathbf{w}$ (e.g. random initialization)

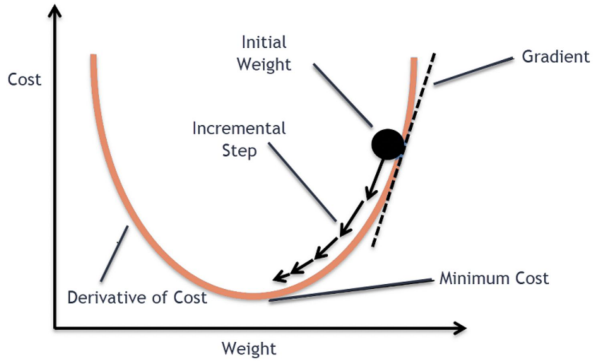
Repeatedly update $\mathbf{w}$ based on the gradient

$$\mathbf{w} \leftarrow \mathbf{w} - \lambda \frac{\partial \ell}{\partial \mathbf{w}}$$

λ is the **learning rate**

Sample: $y(x) = 0.5 + 0.25x$

Sample Update: $y(x) = 0.60 + 0.75x$



Optimizing the Objective

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

Optimization

How to figure out the weights?

$$y(x) = w_0 + w_1 x$$

We **need to find weights** $\mathbf{w}$ such that it minimizes the loss $\ell(\mathbf{w})$

Using Gradient Descent for one example

Initialize $\mathbf{w}$ (e.g. random initialization)

Repeatedly update $\mathbf{w}$ based on the gradient

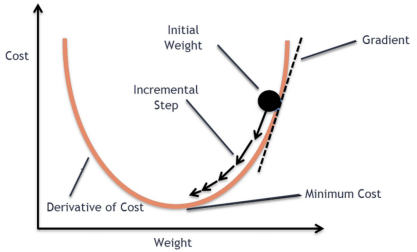
$$\mathbf{w} \leftarrow \mathbf{w} - \lambda \frac{\partial \ell}{\partial \mathbf{w}}$$

λ is the **learning rate**

For a single case, this gives the **least mean squares** update rule:

Sample: $y(x) = 0.5 + 0.25x$

Sample Update: $y(x) = 0.60 + 0.75x$



Optimizing the Objective

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

Optimization

How to figure out the weights?

$$y(x) = w_0 + w_1 x$$

We **need to find weights** $\mathbf{w}$ such that it minimizes the loss $\ell(\mathbf{w})$

Using Gradient Descent for one example

Initialize $\mathbf{w}$ (e.g. random initialization)

Repeatedly update $\mathbf{w}$ based on the gradient

$$\mathbf{w} \leftarrow \mathbf{w} - \lambda \frac{\partial \ell}{\partial \mathbf{w}}$$

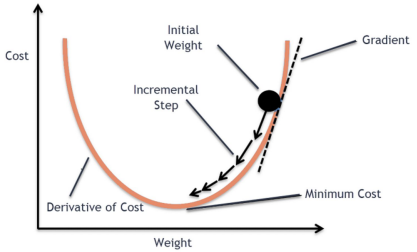
λ is the **learning rate**

For a single case, this gives the **least mean squares** update rule:

$$\mathbf{w} \leftarrow \mathbf{w} + 2\lambda(t^{(n)} - y(x^{(n)}))x^{(n)}$$

Sample: $y(x) = 0.5 + 0.25x$

Sample Update: $y(x) = 0.60 + 0.75x$



Optimizing the Objective

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

- *Features*
- *Training Data*
- *Model*
- *Loss*
- ***Optimization***

How to figure out the weights?

$$y(x) = w_0 + w_1 x$$

We **need to find weights** $\mathbf{w}$ such that it minimizes the loss $\ell(\mathbf{w})$

Using Gradient Descent for one example

Initialize $\mathbf{w}$ (e.g. random initialization)

Sample:

Sample Update:

Repeatedly update $\mathbf{w}$ based on the gradient

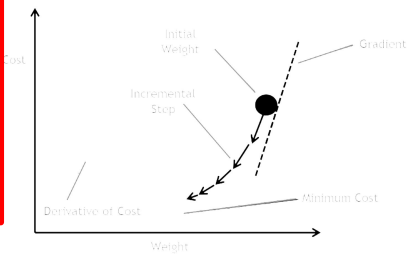
$$\mathbf{w} \leftarrow \mathbf{w} - \lambda \frac{\partial \ell}{\partial \mathbf{w}}$$

λ is the **learning rate**

For a single case, this gives the least mean squares update rule:

After solving the equation, it equates to the following

$$\mathbf{w} \leftarrow \mathbf{w} + 2\lambda(t^{(n)} - y(x^{(n)}))x^{(n)}$$



Optimizing the Objective

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

Optimization

How to figure out the weights?

$$y(x) = w_0 + w_1 x$$

We **need to find weights** $\mathbf{w}$ such that it minimizes the loss $\ell(\mathbf{w})$

Using Gradient Descent for one example

Initialize $\mathbf{w}$ (e.g. random initialization)

Repeatedly update $\mathbf{w}$ based on the gradient

$$\mathbf{w} \leftarrow \mathbf{w} - \lambda \frac{\partial \ell}{\partial \mathbf{w}}$$

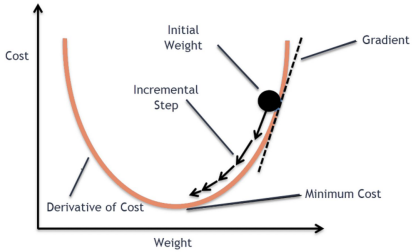
λ is the **learning rate**

For a single case, this gives the **least mean squares** update rule:

$$\mathbf{w} \leftarrow \mathbf{w} + 2\lambda(t^{(n)} - y(x^{(n)}))x^{(n)}$$

Sample: $y(x) = 0.5 + 0.25x$

Sample Update: $y(x) = 0.60 + 0.75x$



Note: *As the error approaches 0, so does the update (* $\mathbf{w}$ *stops changing)*

Optimizing Across the Dataset

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

Optimization

How to optimize across the entire dataset?

Optimizing Across the Dataset

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

Optimization

How to optimize across the entire dataset?

Two ways to generalize this for **all examples in the training set**

Optimizing Across the Dataset

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

Optimization

How to optimize across the entire dataset?

Two ways to generalize this for **all examples in the training set**

1. **Batch updates:** Sum of average updates across every example n, then try to change parameter values

$$\mathbf{w} \leftarrow \mathbf{w} + 2\lambda \sum_{n=1}^{N} (t^{(n)} - y(x^{(n)}))x^{(n)}$$

Optimizing Across the Dataset

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

Optimization

How to optimize across the entire dataset?

Two ways to generalize this for **all examples in the training set**

1. **Batch updates:** Sum of average updates across every example n, then try to change parameter values

$$\mathbf{w} \leftarrow \mathbf{w} + 2\lambda \sum_{n=1}^{N} (t^{(n)} - y(x^{(n)}))x^{(n)}$$

Optimizing Across the Dataset

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

Optimization

How to optimize across the entire dataset?

Two ways to generalize this for **all examples in the training set**

1. **Batch updates:** Sum of average updates across every example n, then try to change parameter values

$$\mathbf{w} \leftarrow \mathbf{w} + 2\lambda \sum_{n=1}^{N} (t^{(n)} - y(x^{(n)}))x^{(n)}$$

2. **Stochastic or Online Updates:** Update the parameters for each training case in turn, according to its own gradients

Optimizing Across the Dataset

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

Optimization

How to optimize across the entire dataset?

Two ways to generalize this for **all examples in the training set**

1. **Batch updates:** Sum of average updates across every example n, then try to change parameter values

$$\mathbf{w} \leftarrow \mathbf{w} + 2\lambda \sum_{n=1}^{N} (t^{(n)} - y(x^{(n)}))x^{(n)}$$

2. **Stochastic or Online Updates:** Update the parameters for each training case in turn, according to its own gradients

Algorithm 1 Stochastic gradient descent

1: Randomly shuffle examples in the training set
2: **for** $i = 1$ to N **do**
3: Update:

$$\mathbf{w} \leftarrow \mathbf{w} + 2\lambda(t^{(i)} - y(x^{(i)}))x^{(i)} \qquad \text{(update for a linear model)}$$

4: **end for**

Optimizing Across the Dataset

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

Optimization

How to optimize across the entire dataset?

Two ways to generalize this for **all examples in the training set**

1. **Batch updates:** Sum of average updates across every example n, then try to change parameter values

$$\mathbf{w} \leftarrow \mathbf{w} + 2\lambda \sum_{n=1}^{N} (t^{(n)} - y(x^{(n)}))x^{(n)}$$

2. **Stochastic or Online Updates:** Update the parameters for each training case in turn, according to its own gradients

Algorithm 1 Stochastic gradient descent

1: Randomly shuffle examples in the training set
2: **for** $i = 1$ to N **do**
3: Update:

$$\mathbf{w} \leftarrow \mathbf{w} + 2\lambda(t^{(i)} - y(x^{(i)}))x^{(i)}$$ (update for a linear model)

4: **end for**

Optimizing Across the Dataset

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

Optimization

How to optimize across the entire dataset?

Two ways to generalize this for **all examples in the training set**

1. **Batch updates:** Sum of average updates across every example n, then try to change parameter values

$$\mathbf{w} \leftarrow \mathbf{w} + 2\lambda \sum_{n=1}^{N} (t^{(n)} - y(x^{(n)}))x^{(n)}$$

2. **Stochastic or Online Updates:** Update the parameters for each training case in turn, according to its own gradients

Algorithm 1 Stochastic gradient descent

1: Randomly shuffle examples in the training set
2: **for** $i = 1$ to N **do**
3: Update:

$$\mathbf{w} \leftarrow \mathbf{w} + 2\lambda(t^{(i)} - y(x^{(i)}))x^{(i)} \qquad \text{(update for a linear model)}$$

4: **end for**

The underlying assumption is that each sample is independent and identically distributed

Visualized Optimization Process

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

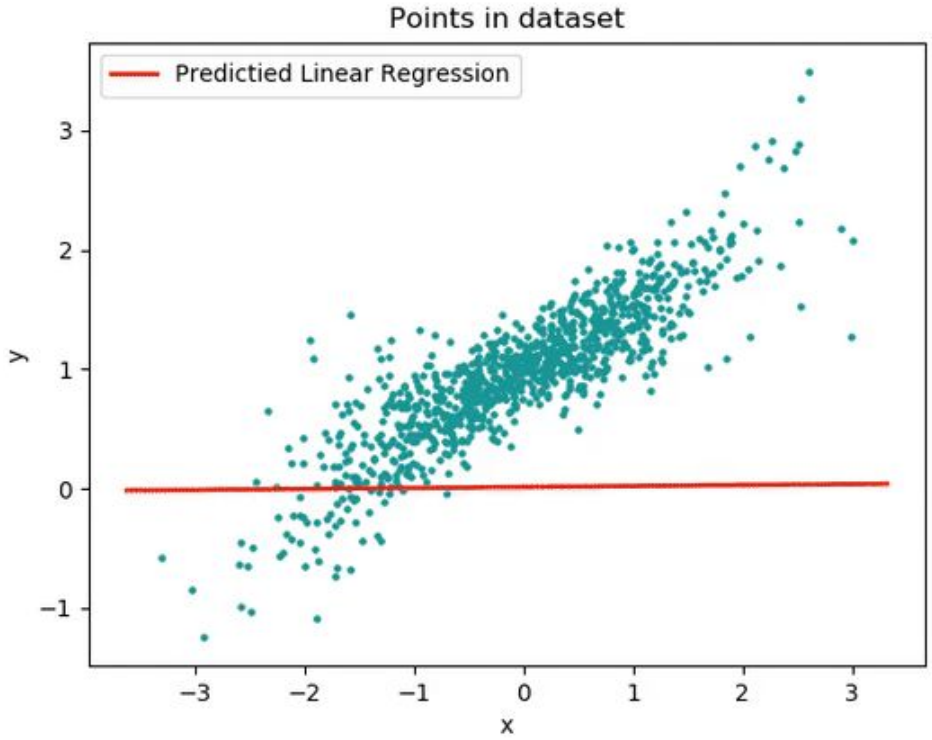
Features

Training Data

Model

Loss

Optimization



Working Example: Boston Housing Data

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Is there a way to improve the model?

Features

Training Data

Model

Loss

Optimization

Working Example: Boston Housing Data

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

- *Features*
- *Training Data*
- *Model*
- *Loss*
- *Optimization*

Is there a way to improve the model?

One method of extending the model is to **consider other input dimensions**

Working Example: Boston Housing Data

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

Optimization

Is there a way to improve the model?

One method of extending the model is to **consider other input dimensions**

$$y(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$$

Working Example: Boston Housing Data

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

Optimization

Is there a way to improve the model?

One method of extending the model is to **consider other input dimensions**

$$y(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$$

x here is a vector now

Working Example: Boston Housing Data

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

Optimization

Is there a way to improve the model?

One method of extending the model is to **consider other input dimensions**

$$y(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$$

In the Boston house pricing example, **number of rooms can also be explored as a feature**

Working Example: Boston Housing Data

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

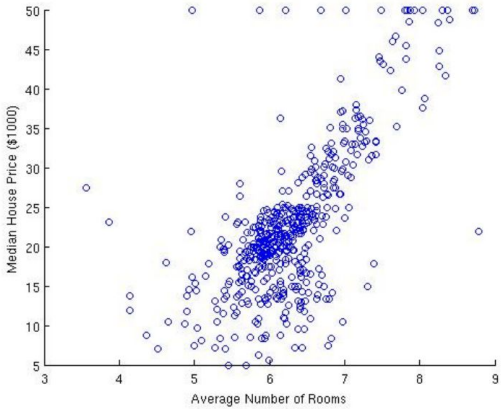
Optimization

Is there a way to improve the model?

One method of extending the model is to **consider other input dimensions**

$$y(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$$

In the Boston house pricing example, **number of rooms can also be explored as a feature**



Working with Multidimensional Inputs

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

Optimization

How do we represent multiple features / multidimensional inputs?

Working with Multidimensional Inputs

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

Optimization

How do we represent multiple features / multidimensional inputs?

Each house is a data point n, with observations indexed by j:

$$\mathbf{x}^{(n)} = \left(x_1^{(n)}, \ldots, x_j^{(n)}, \ldots, x_d^{(n)}\right)$$

Working with Multidimensional Inputs

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

- ***Features***
- *Training Data*
- *Model*
- *Loss*
- *Optimization*

How do we represent multiple features / multidimensional inputs?

Each house is a data point n, with observations indexed by j:

$$\mathbf{x}^{(n)} = \left(x_1^{(n)}, \ldots, x_j^{(n)}, \ldots, x_d^{(n)}\right)$$

We can incorporate the bias w_0 into $\mathbf{w}$, by using $x_0 = 1$ then,

$$y(\mathbf{x}) = w_0 + \sum_{j=1}^{d} w_j x_j = \mathbf{w}^T \mathbf{x}$$

Working with Multidimensional Inputs

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

- ***Features***
- *Training Data*
- *Model*
- *Loss*
- *Optimization*

How do we represent multiple features / multidimensional inputs?

Each house is a data point n, with observations indexed by j:

$$\mathbf{x}^{(n)} = (x_1^{(n)}, \ldots, x_j^{(n)}, \ldots, x_d^{(n)})$$

We can incorporate the bias w_0 into $\mathbf{w}$, by using $x_0 = 1$ then,

$$y(\mathbf{x}) = w_0 + \sum_{j=1}^{d} w_j x_j = \mathbf{w}^T \mathbf{x}$$

$$y(x) = w_0 + w_1 x$$

Basically the same but accounting for more features in the input data

Working with Multidimensional Inputs

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

- ***Features***
- *Training Data*
- *Model*
- *Loss*
- *Optimization*

How do we represent multiple features / multidimensional inputs?

Each house is a data point n, with observations indexed by j:

$$\mathbf{x}^{(n)} = \left(x_1^{(n)}, \ldots, x_j^{(n)}, \ldots, x_d^{(n)}\right)$$

We can incorporate the bias w_0 into $\mathbf{w}$, by using $x_0 = 1$ then,

$$y(\mathbf{x}) = w_0 + \sum_{j=1}^{d} w_j x_j = \mathbf{w}^T \mathbf{x}$$

We can then solve for $\mathbf{w} = (w_0, w_1, \ldots, w_d)$. How?

Working with Multidimensional Inputs

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

Optimization

How do we represent multiple features / multidimensional inputs?

Each house is a data point n, with observations indexed by j:

$$\mathbf{x}^{(n)} = \left(x_1^{(n)}, \ldots, x_j^{(n)}, \ldots, x_d^{(n)}\right)$$

We can incorporate the bias w_0 into $\mathbf{w}$, by using $x_0 = 1$ then,

$$y(\mathbf{x}) = w_0 + \sum_{j=1}^{d} w_j x_j = \mathbf{w}^T \mathbf{x}$$

We can then solve for $\mathbf{w} = (w_0, w_1, \ldots, w_d)$. How?

We can **use gradient descent to solve for each coefficient**, or compute $\mathbf{w}$ analytically.

Increasing Complexity

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

What if the linear model is not good? Can we create a more complicated model?

Features

Training Data

Model

Loss

Optimization

Increasing Complexity

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

Optimization

What if the linear model is not good? Can we create a more complicated model?

We can create a more complicated model by **defining input variables that are combinations of components** of $\mathbf{x}$

Increasing Complexity

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

Optimization

What if the linear model is not good? Can we create a more complicated model?

We can create a more complicated model by **defining input variables that are combinations of components** of $\mathbf{x}$

An M-th order polynomial function one dimensional feature x:

Increasing Complexity

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

Optimization

What if the linear model is not good? Can we create a more complicated model?

We can create a more complicated model by **defining input variables that are combinations of components** of $\mathbf{x}$

An M-th order polynomial function one dimensional feature x:

$$y(x, \mathbf{w}) = w_0 + \sum_{j=1}^{M} w_j x^j$$

where x^j is the j-th power of x

Increasing Complexity

Objective: **Estimate median house price in a neighborhood** based on neighborhood statistics

Features

Training Data

Model

Loss

Optimization

What if the linear model is not good? Can we create a more complicated model?

We can create a more complicated model by **defining input variables that are combinations of components** of $\mathbf{x}$

An M-th order polynomial function one dimensional feature x:

$$y(x, \mathbf{w}) = w_0 + \sum_{j=1}^{M} w_j x^j$$ where x^j is the j-th power of x

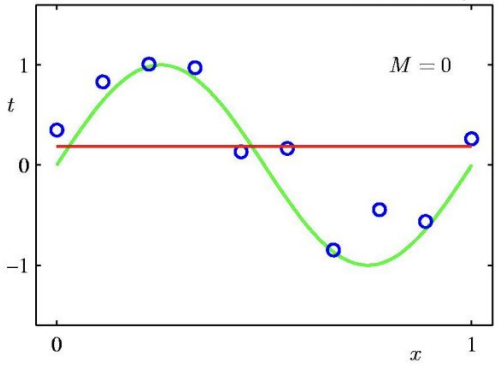
We can use the **same approach to optimize** for the weights $\mathbf{w}$

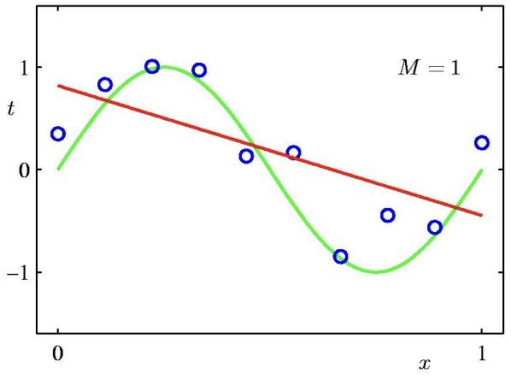
Increasing Complexity

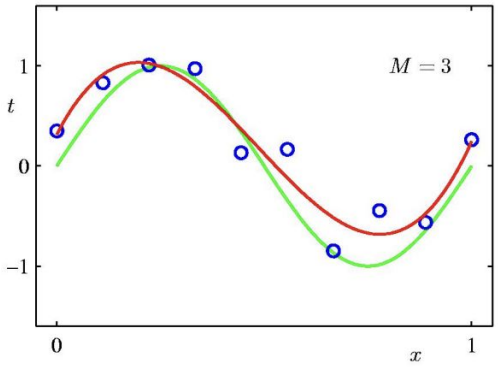
Which fit is the best?

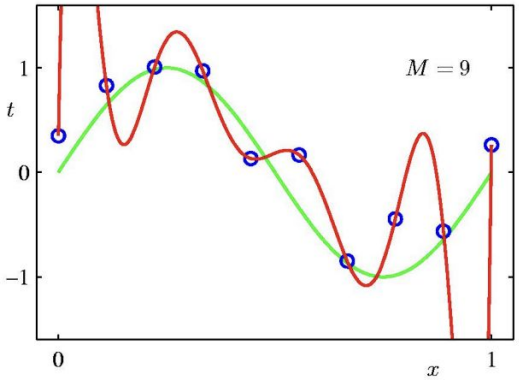
- *Features*
- *Training Data*
- ***Model***
- *Loss*
- *Optimization*

from Bishop









Generalization

What is generalization?

Features

Training Data

Model

Loss

Optimization

Generalization

What is generalization?

Features

Training Data

Model

Loss

Optimization

Generalization is the **model's ability to predict the held out data**

Generalization

What is generalization?

Features

Training Data

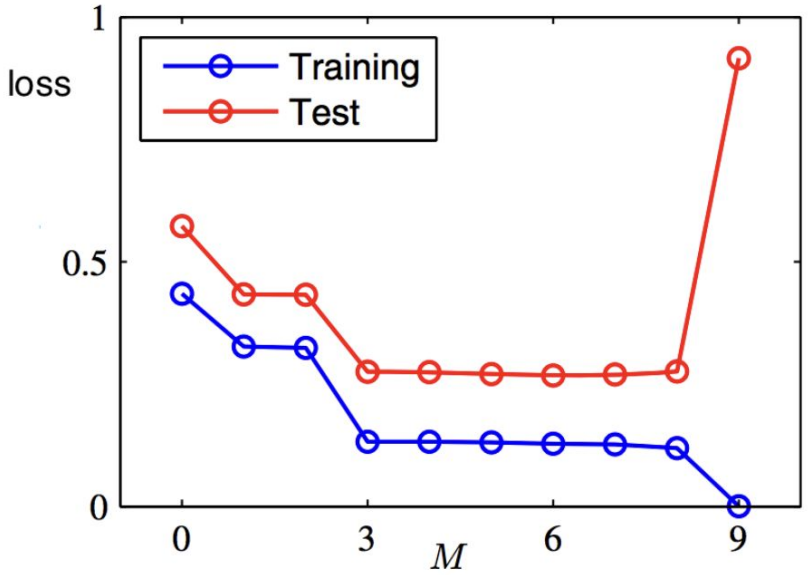
Model

Loss

Optimization

Generalization is the **model's ability to predict the held out data**

What is happening?



Generalization

What is generalization?

Features

Training Data

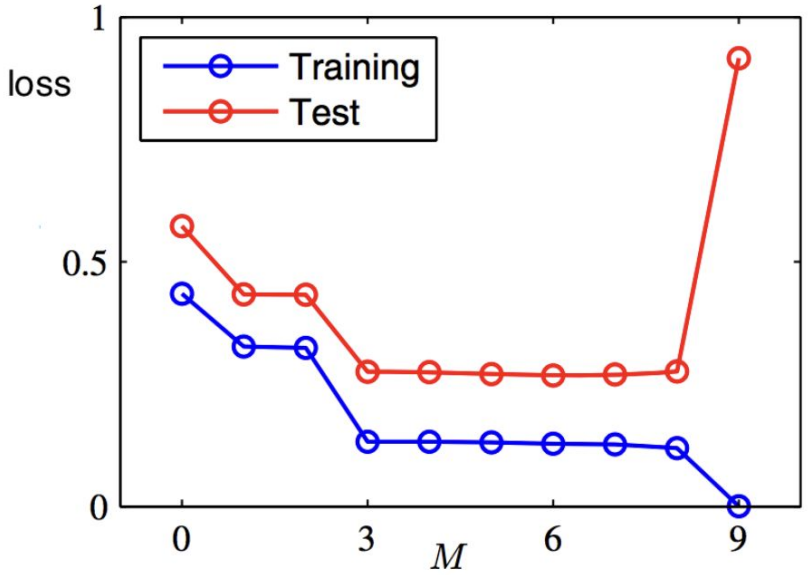
Model

Loss

Optimization

Generalization is the **model's ability to predict the held out data**

What is happening? Our model with M = 9 **overfits the data** (it models also noise)



Generalization

What is generalization?

Features

Training Data

Model

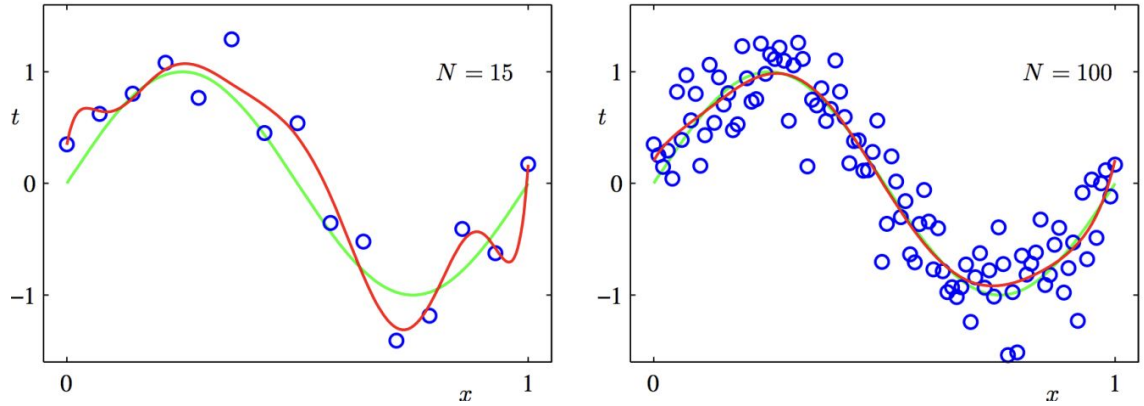
Loss

Optimization

Generalization is the **model's ability to predict the held out data**

What is happening? Our model with M = 9 **overfits the data** (it models also noise)

Not a problem if we have **lots of training examples**



Generalization

What is generalization?

Features

Training Data

Model

Loss

Optimization

Generalization is the **model's ability to predict the held out data**

What is happening? Our model with M = 9 **overfits the data** (it models also noise)

Not a problem if we have **lots of training examples**

Let's look at the **estimated weights** for various M in the case of fewer examples

	$M=0$	$M=1$	$M=6$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

Generalization

What is generalization?

Features

Training Data

Model

Loss

Optimization

Generalization is the **model's ability to predict the held out data**

What is happening? Our model with M = 9 **overfits the data** (it models also noise)

Not a problem if we have **lots of training examples**

The **weights are becoming huge to compensate** for the noise

	$M=0$	$M=1$	$M=6$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

Generalization

What is generalization?

Features

Training Data

Model

Loss

Optimization

Generalization is the **model's ability to predict the held out data**

What is happening? Our model with $M = 9$ **overfits the data** (it models also noise)

Not a problem if we have **lots of training examples**

The **weights are becoming huge to compensate** for the noise

One way of dealing with this is to **encourage the weights to be small** (this way no input dimension will have too much influence on prediction). This is called **regularization**.

Regularization

How to regularize?

- *Features*
- *Training Data*
- *Model*
- ***Loss***
- *Optimization*

Regularization

How to regularize?

Features

Training Data

Model

Loss

Optimization

Goal: Select the appropriate model complexity automatically

Regularization

How to regularize?

- *Features*
- *Training Data*
- *Model*
- ***Loss***
- *Optimization*

Goal: Select the appropriate model complexity automatically

Standard approach: **Regularization**

$$\tilde{\ell} = \sum_{n=1}^{N} \left[t^{(n)} - (w_0 + w_1 x^{(n)}) \right]^2 + \alpha \mathbf{w}^T \mathbf{w}$$

Regularization

How to regularize?

- *Features*
- *Training Data*
- *Model*
- ***Loss***
- *Optimization*

Goal: Select the appropriate model complexity automatically

Standard approach: **Regularization**

$$\tilde{\ell} = \sum_{n=1}^{N} \left[t^{(n)} - (w_0 + w_1 x^{(n)}) \right]^2 + \alpha \mathbf{w}^T \mathbf{w}$$

Penalty term

Regularization

How to regularize?

- *Features*
- *Training Data*
- *Model*
- ***Loss***
- *Optimization*

Goal: Select the appropriate model complexity automatically

Standard approach: **Regularization**

$$\tilde{\ell} = \sum_{n=1}^{N} \left[t^{(n)} - (w_0 + w_1 x^{(n)}) \right]^2 + \alpha \mathbf{w}^T \mathbf{w}$$

Intuition: Since we are minimizing the loss, the second term will encourage smaller values in

Regularization

How to regularize?

- *Features*
- *Training Data*
- *Model*
- ***Loss***
- *Optimization*

Goal: Select the appropriate model complexity automatically

Standard approach: **Regularization**

$$\tilde{\ell} = \sum_{n=1}^{N} \left[t^{(n)} - (w_0 + w_1 x^{(n)}) \right]^2 + \alpha \mathbf{w}^T \mathbf{w}$$

Intuition: Since we are minimizing the loss, the second term will encourage smaller values in

The penalty on the squared weights is known as **ridge regression in statistics**

Regularization

How to regularize?

Features

Training Data

Model

Loss

Optimization

Goal: Select the appropriate model complexity automatically

Standard approach: **Regularization**

$$\tilde{\ell} = \sum_{n=1}^{N} \left[t^{(n)} - (w_0 + w_1 x^{(n)}) \right]^2 + \alpha \mathbf{w}^T \mathbf{w}$$

Intuition: Since we are minimizing the loss, the second term will encourage smaller values in

The penalty on the squared weights is known as **ridge regression in statistics**

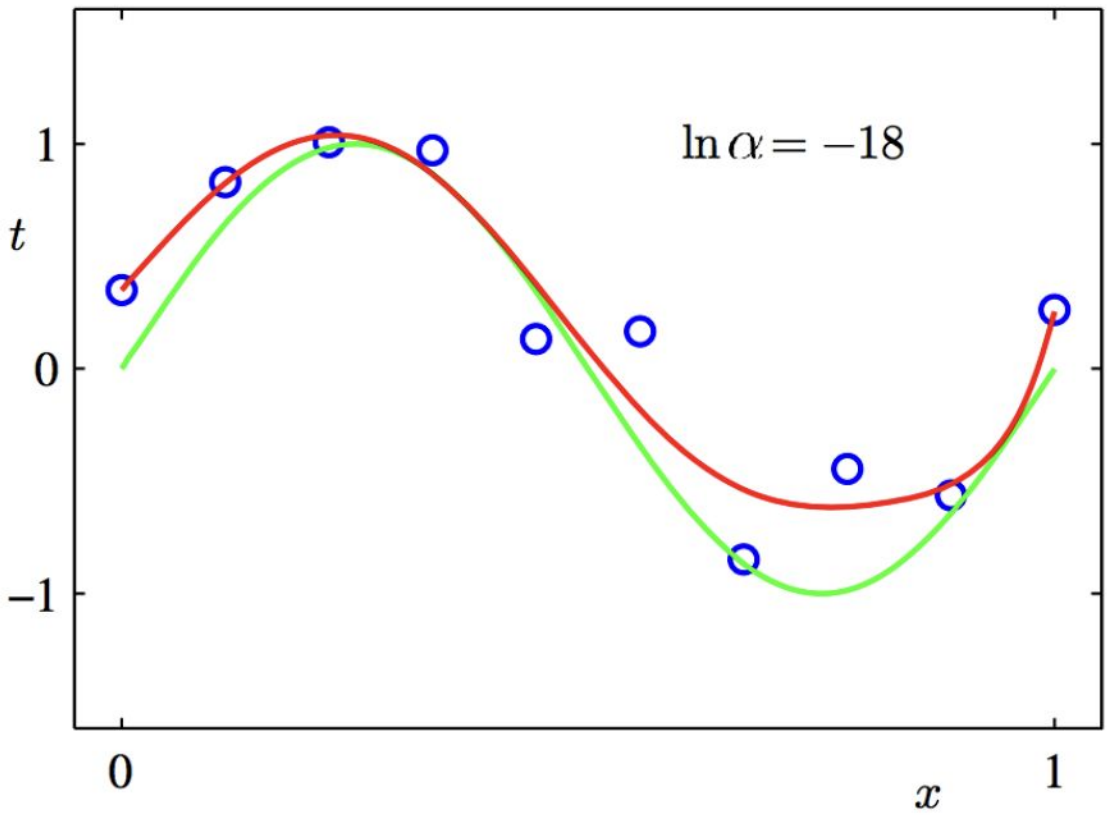
Leads to a **modified update rule** for gradient descent:

$$\mathbf{w} \leftarrow \mathbf{w} + 2\lambda \left[\sum_{n=1}^{N} (t^{(n)} - y(x^{(n)})) x^{(n)} - \alpha \mathbf{w} \right]$$

Regularization

Choose α carefully

- *Features*
- *Training Data*
- *Model*
- ***Loss***
- *Optimization*



In Summary

What have we learned for today?

- ***Features***
- *Training Data*
- *Model*
- *Loss*
- *Optimization*

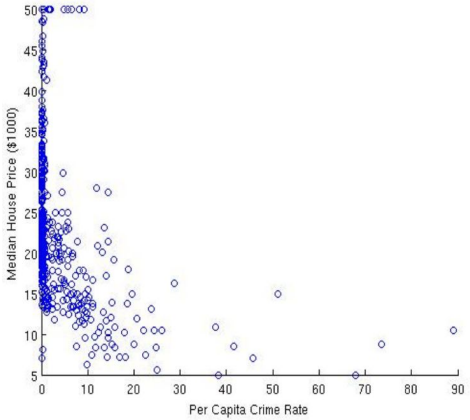
Features: **Per Capita Crime Rate** and **Average Number of Rooms**

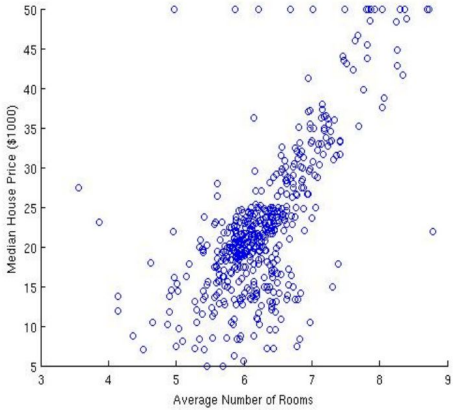
In Summary

What have we learned for today?

- ***Features***
- *Training Data*
- *Model*
- *Loss*
- *Optimization*

Features: **Per Capita Crime Rate** and **Average Number of Rooms**





In Summary

What have we learned for today?

- *Features*
- ***Training Data***
- *Model*
- *Loss*
- *Optimization*

Features: **Per Capita Crime Rate** and **Average Number of Rooms**

***Training* Data Structure**

$$D = \{(\mathbf{x}^{(1)}, t^{(1)}), \ldots, (\mathbf{x}^{(n)}, t^{(n)})\}$$

In Summary

What have we learned for today?

- *Features*
- ***Training Data***
- *Model*
- *Loss*
- *Optimization*

Features: **Per Capita Crime Rate** and **Average Number of Rooms**

***Training* Data Structure**

$$D = \{(\mathbf{x}^{(1)}, t^{(1)}), \ldots, (\mathbf{x}^{(n)}, t^{(n)})\}$$

such that

In Summary

What have we learned for today?

- *Features*
- ***Training Data***
- *Model*
- *Loss*
- *Optimization*

Features: **Per Capita Crime Rate** and **Average Number of Rooms**

***Training* Data Structure**

$$D = \{(\mathbf{x}^{(1)}, t^{(1)}), \ldots, (\mathbf{x}^{(n)}, t^{(n)})\}$$

such that

$$\mathbf{x} = (x_1^{(n)}, x_2^{(n)})$$

In Summary

What have we learned for today?

- *Features*
- ***Training Data***
- *Model*
- *Loss*
- *Optimization*

Features: **Per Capita Crime Rate** and **Average Number of Rooms**

***Training* Data Structure**

$$D = \{(\mathbf{x}^{(1)}, t^{(1)}), \ldots, (\mathbf{x}^{(n)}, t^{(n)})\}$$

such that

$$\mathbf{x} = (x_1^{(n)}, x_2^{(n)})$$

$$x_1 = \text{per capita crime rate}$$

In Summary

What have we learned for today?

- *Features*
- ***Training Data***
- *Model*
- *Loss*
- *Optimization*

Features: **Per Capita Crime Rate** and **Average Number of Rooms**

***Training* Data Structure**

$$D = \{(\mathbf{x}^{(1)}, t^{(1)}), \ldots, (\mathbf{x}^{(n)}, t^{(n)})\}$$

such that

$$\mathbf{x} = (x_1^{(n)}, x_2^{(n)})$$

$$x_1 = \text{per capita crime rate}$$

$$x_2 = \text{number of rooms}$$

In Summary

What have we learned for today?

- *Features*
- ***Training Data***
- *Model*
- *Loss*
- *Optimization*

Features: **Per Capita Crime Rate** and **Average Number of Rooms**

***Training* Data Structure**

$$D = \{(\mathbf{x}^{(1)}, t^{(1)}), \ldots, (\mathbf{x}^{(n)}, t^{(n)})\}$$

such that

$$\mathbf{x} = (x_1^{(n)}, x_2^{(n)})$$

$$x_1 = \text{per capita crime rate}$$
$$x_2 = \text{number of rooms}$$
$$t = \text{median house price}$$

In Summary

What have we learned for today?

- *Features*
- ***Training Data***
- *Model*
- *Loss*
- *Optimization*

Features: **Per Capita Crime Rate** and **Average Number of Rooms**

***Training* Data Structure**

$$D = \{(\mathbf{x}^{(1)}, t^{(1)}), \ldots, (\mathbf{x}^{(n)}, t^{(n)})\}$$

such that

$$\mathbf{x} = (x_1^{(n)}, x_2^{(n)})$$

$x_1 = \text{per capita crime rate}$
$x_2 = \text{number of rooms}$
$t = \text{median house price}$
$n = \text{number of data points}$

In Summary

What have we learned for today?

Features

Training Data

Model

Loss

Optimization

Features: **Per Capita Crime Rate** and **Average Number of Rooms**

***Training* Data Structure**

$$D = \{(\mathbf{x}^{(1)}, t^{(1)}), \ldots, (\mathbf{x}^{(n)}, t^{(n)})\}$$

such that

$$\mathbf{x} = (x_1^{(n)}, x_2^{(n)})$$

$x_1 =$ per capita crime rate
$x_2 =$ number of rooms
$t =$ median house price
$n =$ number of data points

Sample Data:

Neighborhood #1

$x_1 = 40 \quad x_2 = 3 \quad t = 3,400$

Neighborhood #2

$x_1 = 20 \quad x_2 = 5 \quad t = 4,500$

Neighborhood #3

$x_1 = 30 \quad x_2 = 2 \quad t = 2,800$

In Summary

What have we learned for today?

- *Features*
- *Training Data*
- ***Model***
- *Loss*
- *Optimization*

Features: **Per Capita Crime Rate** and **Average Number of Rooms**

***Training* Data Structure**

$$D = \{(\mathbf{x}^{(1)}, t^{(1)}), \ldots, (\mathbf{x}^{(n)}, t^{(n)})\}$$

Multivariate Linear Regression

$$y(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$$

In Summary

What have we learned for today?

- *Features*
- *Training Data*
- *Model*
- ***Loss***
- *Optimization*

Features: **Per Capita Crime Rate** and **Average Number of Rooms**

***Training* Data Structure**

$$D = \{(\mathbf{x}^{(1)}, t^{(1)}), \ldots, (\mathbf{x}^{(n)}, t^{(n)})\}$$

Multivariate Linear Regression

$$y(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$$

***Sum of Squares* Error**

$$\ell(\mathbf{w}) = \sum_{n=1}^{N} [t^{(n)} - (w_0 + w_1 x^{(n)})]^2$$

In Summary

What have we learned for today?

- *Features*
- *Training Data*
- *Model*
- *Loss*
- ***Optimization***

Features: **Per Capita Crime Rate** and **Average Number of Rooms**

***Training* Data Structure**

$$D = \{(\mathbf{x}^{(1)}, t^{(1)}), \ldots, (\mathbf{x}^{(n)}, t^{(n)})\}$$

Multivariate Linear Regression

$$y(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$$

***Sum of Squares* Error**

$$\ell(\mathbf{w}) = \sum_{n=1}^{N} [t^{(n)} - (w_0 + w_1 x^{(n)})]^2$$

Optimization using Gradient Descent

$$\mathbf{w} \leftarrow \mathbf{w} + 2\lambda \sum_{n=1}^{N} (t^{(n)} - y(x^{(n)}))x^{(n)}$$

In Summary

What have we learned for today?

Multivariate Linear Regression $y(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$

- *Features*
- *Training Data*
- *Model*
- *Loss*
- ***Optimization***

Sample Data:

Neighborhood #1

$x_1 = 40 \quad x_2 = 3 \quad t = 3,400$

Neighborhood #2

$x_1 = 20 \quad x_2 = 5 \quad t = 4,500$

Neighborhood #3

$x_1 = 30 \quad x_2 = 2 \quad t = 2,800$

In Summary

What have we learned for today?

- Features
- Training Data
- Model
- Loss
- **Optimization**

Multivariate Linear Regression $y(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$

Gradient Descent

Sample Data:

Neighborhood #1

$x_1 = 40 \quad x_2 = 3 \quad t = 3,400$

Neighborhood #2

$x_1 = 20 \quad x_2 = 5 \quad t = 4,500$

Neighborhood #3

$x_1 = 30 \quad x_2 = 2 \quad t = 2,800$

In Summary

What have we learned for today?

Features

Training Data

Model

Loss

Optimization

Multivariate Linear Regression $y(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$

Gradient Descent

Step #1: Initialize Weights

Sample Data:

Neighborhood #1

$x_1 = 40 \quad x_2 = 3 \quad t = 3,400$

Neighborhood #2

$x_1 = 20 \quad x_2 = 5 \quad t = 4,500$

Neighborhood #3

$x_1 = 30 \quad x_2 = 2 \quad t = 2,800$

In Summary

What have we learned for today?

- Features
- Training Data
- Model
- Loss
- **Optimization**

Multivariate Linear Regression $y(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$

Gradient Descent

Step #1: Initialize Weights

$$y(\mathbf{x}) = 200 + 5(x_1) + 200\,(x_2)$$

Sample Data:

Neighborhood #1

$x_1 = 40 \quad x_2 = 3 \quad t = 3,400$

Neighborhood #2

$x_1 = 20 \quad x_2 = 5 \quad t = 4,500$

Neighborhood #3

$x_1 = 30 \quad x_2 = 2 \quad t = 2,800$

In Summary

What have we learned for today?

- *Features*
- *Training Data*
- *Model*
- *Loss*
- ***Optimization***

Multivariate Linear Regression $y(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$

Gradient Descent

Step #1: Initialize Weights

$$y(\mathbf{x}) = 200 + 5(x_1) + 200\,(x_2)$$

Step #2: Update Weights

$$\mathbf{w} \leftarrow \mathbf{w} - \lambda \frac{\partial \ell}{\partial \mathbf{w}} \qquad \mathbf{w} \leftarrow \mathbf{w} + 2\lambda(t^{(n)} - y(x^{(n)}))x^{(n)}$$

Sample Data:

Neighborhood #1

$x_1 = 40 \quad x_2 = 3 \quad t = 3,400$

Neighborhood #2

$x_1 = 20 \quad x_2 = 5 \quad t = 4,500$

Neighborhood #3

$x_1 = 30 \quad x_2 = 2 \quad t = 2,800$

In Summary

What have we learned for today?

- *Features*
- *Training Data*
- *Model*
- *Loss*
- ***Optimization***

Multivariate Linear Regression $y(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$

Gradient Descent

Step #1: Initialize Weights

$$y(\mathbf{x}) = 200 + 5(x_1) + 200\,(x_2)$$

Step #2: Update Weights

$$\mathbf{w} \leftarrow \mathbf{w} - \lambda \frac{\partial \ell}{\partial \mathbf{w}} \qquad \mathbf{w} \leftarrow \mathbf{w} + 2\lambda(t^{(n)} - y(x^{(n)}))x^{(n)}$$

Neighborhood #1

Sample Data:

Neighborhood #1

$x_1 = 40 \quad x_2 = 3 \quad t = 3,400$

Neighborhood #2

$x_1 = 20 \quad x_2 = 5 \quad t = 4,500$

Neighborhood #3

$x_1 = 30 \quad x_2 = 2 \quad t = 2,800$

In Summary

What have we learned for today?

Features

Training Data

Model

Loss

Optimization

Multivariate Linear Regression $y(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$

Gradient Descent

Step #1: Initialize Weights

$$y(\mathbf{x}) = 200 + 5(x_1) + 200\,(x_2)$$

Step #2: Update Weights

$$\mathbf{w} \leftarrow \mathbf{w} - \lambda \frac{\partial \ell}{\partial \mathbf{w}} \qquad \mathbf{w} \leftarrow \mathbf{w} + 2\lambda(t^{(n)} - y(x^{(n)}))x^{(n)}$$

Neighborhood #1

$$w_0 = 200 + 2(1)(3400 - (200 + 5(40) + 200(3)))1$$

Sample Data:

Neighborhood #1

$x_1 = 40 \quad x_2 = 3 \quad t = 3,400$

Neighborhood #2

$x_1 = 20 \quad x_2 = 5 \quad t = 4,500$

Neighborhood #3

$x_1 = 30 \quad x_2 = 2 \quad t = 2,800$

In Summary

What have we learned for today?

Features

Training Data

Model

Loss

Optimization

Multivariate Linear Regression $y(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$

Gradient Descent

Step #1: Initialize Weights

$$y(\mathbf{x}) = 200 + 5(x_1) + 200\,(x_2)$$

Step #2: Update Weights

$$\mathbf{w} \leftarrow \mathbf{w} - \lambda \frac{\partial \ell}{\partial \mathbf{w}} \qquad \mathbf{w} \leftarrow \mathbf{w} + 2\lambda(t^{(n)} - y(x^{(n)}))x^{(n)}$$

Neighborhood #1

$$w_0 = 200 + 2(1)(3400 - (200 + 5(40) + 200(3)))1$$

$$w_1 = 5 + 2(1)(3400 - (200 + 5(40) + 200(3)))40$$

Sample Data:

Neighborhood #1

$x_1 = 40 \quad x_2 = 3 \quad t = 3,400$

Neighborhood #2

$x_1 = 20 \quad x_2 = 5 \quad t = 4,500$

Neighborhood #3

$x_1 = 30 \quad x_2 = 2 \quad t = 2,800$

In Summary

What have we learned for today?

Features

Training Data

Model

Loss

Optimization

Multivariate Linear Regression $y(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$

Gradient Descent

Step #1: Initialize Weights

$$y(\mathbf{x}) = 200 + 5(x_1) + 200\,(x_2)$$

Step #2: Update Weights

$$\mathbf{w} \leftarrow \mathbf{w} - \lambda \frac{\partial \ell}{\partial \mathbf{w}} \qquad \mathbf{w} \leftarrow \mathbf{w} + 2\lambda(t^{(n)} - y(x^{(n)}))x^{(n)}$$

Neighborhood #1

$$w_0 = 200 + 2(1)(3400 - (200 + 5(40) + 200(3)))1$$

$$w_1 = 5 + 2(1)(3400 - (200 + 5(40) + 200(3)))40$$

$$w_2 = 200 + 2(1)(3400 - (200 + 5(40) + 200(3)))3$$

Sample Data:

Neighborhood #1

$x_1 = 40 \quad x_2 = 3 \quad t = 3,400$

Neighborhood #2

$x_1 = 20 \quad x_2 = 5 \quad t = 4,500$

Neighborhood #3

$x_1 = 30 \quad x_2 = 2 \quad t = 2,800$

In Summary

What have we learned for today?

- *Features*
- *Training Data*
- *Model*
- *Loss*
- ***Optimization***

Multivariate Linear Regression $y(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$

Gradient Descent

Step #1: Initialize Weights

$$y(\mathbf{x}) = 200 + 5(x_1) + 200\,(x_2)$$

Step #2: Update Weights

$$\mathbf{w} \leftarrow \mathbf{w} - \lambda \frac{\partial \ell}{\partial \mathbf{w}} \qquad \mathbf{w} \leftarrow \mathbf{w} + 2\lambda(t^{(n)} - y(x^{(n)}))x^{(n)}$$

Neighborhood #1

$$w_0 = 200 + 2(1)(3400 - (200 + 5(40) + 200(3)))1 \qquad w_0 = 5,000$$

$$w_1 = 5 + 2(1)(3400 - (200 + 5(40) + 200(3)))40 \qquad w_1 = 192,005$$

$$w_2 = 200 + 2(1)(3400 - (200 + 5(40) + 200(3)))3 \qquad w_2 = 15,400$$

Sample Data:

Neighborhood #1

$x_1 = 40 \quad x_2 = 3 \quad t = 3,400$

Neighborhood #2

$x_1 = 20 \quad x_2 = 5 \quad t = 4,500$

Neighborhood #3

$x_1 = 30 \quad x_2 = 2 \quad t = 2,800$

In Summary

What have we learned for today?

- *Features*
- *Training Data*
- *Model*
- *Loss*
- ***Optimization***

Multivariate Linear Regression $y(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$

Gradient Descent

Step #1: Initialize Weights

$$y(\mathbf{x}) = 200 + 5(x_1) + 200\,(x_2)$$

Step #2: Update Weights

$$\mathbf{w} \leftarrow \mathbf{w} - \lambda \frac{\partial \ell}{\partial \mathbf{w}} \qquad \mathbf{w} \leftarrow \mathbf{w} + 2\lambda (t^{(n)} - y(x^{(n)}))x^{(n)}$$

Neighborhood #2

$$w_0 = 5000 + 2(1)(4500 - (5000 + 192005(20) + 15400(5)))1$$

Sample Data:

Neighborhood #1

$x_1 = 40 \quad x_2 = 3 \quad t = 3,400$

Neighborhood #2

$x_1 = 20 \quad x_2 = 5 \quad t = 4,500$

Neighborhood #3

$x_1 = 30 \quad x_2 = 2 \quad t = 2,800$

In Summary

What have we learned for today?

- *Features*
- *Training Data*
- *Model*
- *Loss*
- ***Optimization***

Multivariate Linear Regression $y(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$

Gradient Descent

Step #1: Initialize Weights

$$y(\mathbf{x}) = 200 + 5(x_1) + 200\,(x_2)$$

Step #2: Update Weights

$$\mathbf{w} \leftarrow \mathbf{w} - \lambda \frac{\partial \ell}{\partial \mathbf{w}} \qquad \mathbf{w} \leftarrow \mathbf{w} + 2\lambda(t^{(n)} - y(x^{(n)}))x^{(n)}$$

Neighborhood #2

$$w_0 = 5000 + 2(1)(4500 - (5000 + 192005(20) + 15400(5)))1$$

$$w_1 = 192005 + 2(1)(4500 - (5000 + 192005(20) + 15400(5)))20$$

Sample Data:

Neighborhood #1

$x_1 = 40 \quad x_2 = 3 \quad t = 3,400$

Neighborhood #2

$x_1 = 20 \quad x_2 = 5 \quad t = 4,500$

Neighborhood #3

$x_1 = 30 \quad x_2 = 2 \quad t = 2,800$

In Summary

What have we learned for today?

Features

Training Data

Model

Loss

Optimization

Multivariate Linear Regression $y(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$

Gradient Descent

Step #1: Initialize Weights

$$y(\mathbf{x}) = 200 + 5(x_1) + 200\,(x_2)$$

Step #2: Update Weights

$$\mathbf{w} \leftarrow \mathbf{w} - \lambda \frac{\partial \ell}{\partial \mathbf{w}} \qquad \mathbf{w} \leftarrow \mathbf{w} + 2\lambda(t^{(n)} - y(x^{(n)}))x^{(n)}$$

Neighborhood #2

$$w_0 = 5000 + 2(1)(4500 - (5000 + 192005(20) + 15400(5)))1$$
$$w_1 = 192005 + 2(1)(4500 - (5000 + 192005(20) + 15400(5)))20$$
$$w_2 = 15400 + 2(1)(4500 - (5000 + 192005(20) + 15400(5)))5$$

Sample Data:

Neighborhood #1

$x_1 = 40 \quad x_2 = 3 \quad t = 3,400$

Neighborhood #2

$x_1 = 20 \quad x_2 = 5 \quad t = 4,500$

Neighborhood #3

$x_1 = 30 \quad x_2 = 2 \quad t = 2,800$

In Summary

What have we learned for today?

Features

Training Data

Model

Loss

Optimization

Multivariate Linear Regression $y(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$

Gradient Descent

Step #1: Initialize Weights

$$y(\mathbf{x}) = 200 + 5(x_1) + 200\,(x_2)$$

Step #2: Update Weights

$$\mathbf{w} \leftarrow \mathbf{w} - \lambda \frac{\partial \ell}{\partial \mathbf{w}} \qquad \mathbf{w} \leftarrow \mathbf{w} + 2\lambda(t^{(n)} - y(x^{(n)}))x^{(n)}$$

Neighborhood #2

$$w_0 = 5000 + 2(1)(4500 - (5000 + 192005(20) + 15400(5)))1 \qquad w_0 = -7,830,200$$

$$w_1 = 192005 + 2(1)(4500 - (5000 + 192005(20) + 15400(5)))20 \qquad w_1 = -156,551,995$$

$$w_2 = 15400 + 2(1)(4500 - (5000 + 192005(20) + 15400(5)))5 \qquad w_2 = -36,160,600$$

Sample Data:

Neighborhood #1

$x_1 = 40 \quad x_2 = 3 \quad t = 3,400$

Neighborhood #2

$x_1 = 20 \quad x_2 = 5 \quad t = 4,500$

Neighborhood #3

$x_1 = 30 \quad x_2 = 2 \quad t = 2,800$

In Summary

What have we learned for today?

- ***Features***
- ***Training Data***
- ***Model***
- ***Loss***
- ***Optimization***

We have learned…

What are Loss Functions and how Linear Regression is done

We have identified the necessary components for linear regression

Features, Training Data, Model, Loss, and Optimization

We discussed concepts about…

Noie Generalization, Regularization, etc.