Machine Learning (Lecture 1)

UEM/IEM Summer 2018

Why Use Machine Learning (ML)?

- To do three practical things better as a (software) engineer:
    1. Reduce time programming
    2. Customize and scale products
    3. Complete seemingly “unprogrammable” tasks

Why Use Machine Learning (ML)?

- Philosophical reasons:
  - ML changes the way you think about problems.
    - Software engineers think logically and mathematically
    - Focus shift in ML:
      - Mathematical science to natural science
      - Observations of uncertain world
      - Running experiments
      - Use statistics (not logic) to analyze the experiments
      - Think like scientists
      - Open up new areas to explore with ML

What is (Supervised) ML?

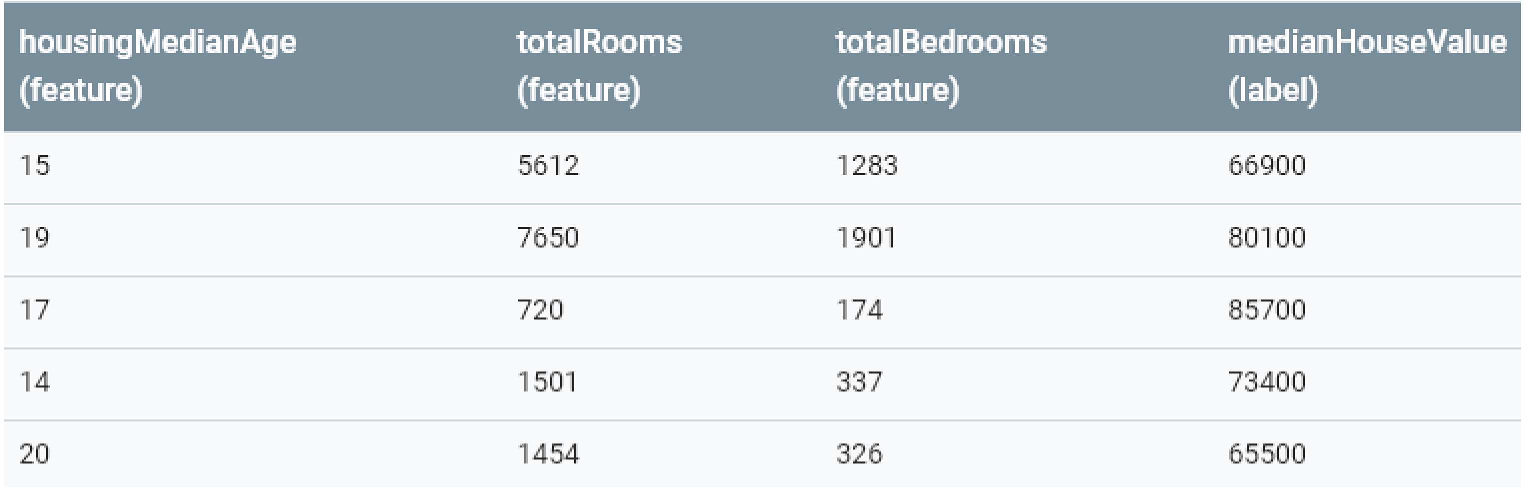
- ML systems learn how to combine input to produce useful predictions on never-before-seen data

What is (Supervised) ML?

- Terminology: Labels and Features
  - Label is the true thing we’re predicting: y
    - The y variable in basic linear regression
    - The label could be the future price of wheat, the kind of animal shown in a picture, the meaning of an audio clip, or just about anything.
  - Features are input variables describing our data: x_i
    - The $\{x_1, x_2, \ldots x_n\}$ variables in basic linear regression
    - A simple machine learning project might use a single feature, while a more sophisticated machine learning project could use millions of features.
    - In the spam detector example, the features could include the following:
      - words in the email text
      - sender's address
      - time of day the email was sent
      - email contains the phrase "one weird trick."

What is (Supervised) ML?

- Terminology: Examples
  - Example is a particular instance of data, **x**
    (**x** is a vector)
  - Labeled example has {features, label}: (x, y)
    - Used to **train** the model

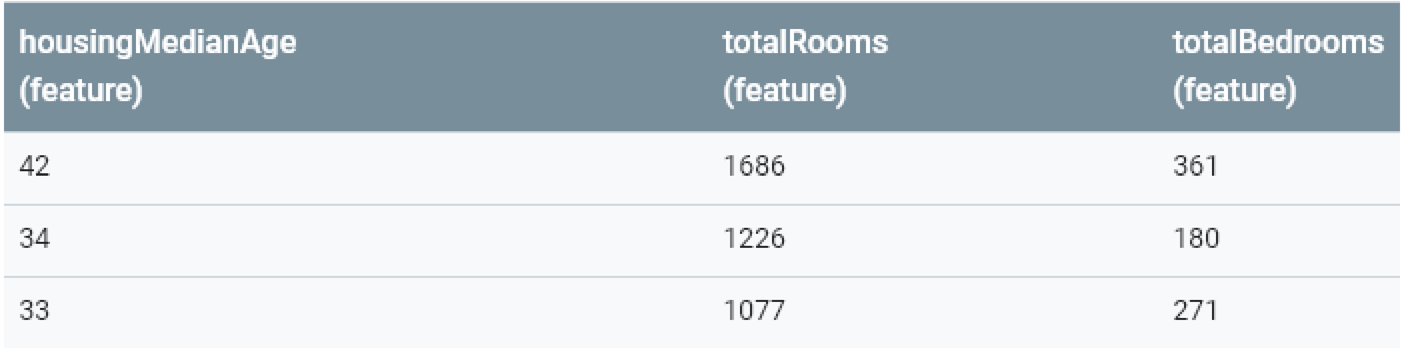


housingMedianAge (feature)	totalRooms (feature)	totalBedrooms (feature)	medianHouseValue (label)
15	5612	1283	66900
19	7650	1901	80100
17	720	174	85700
14	1501	337	73400
20	1454	326	65500

    - In our spam detector example, the labeled examples would be individual emails that users have explicitly marked as "spam" or "not spam."

What is (Supervised) ML?

- Terminology: Examples
  - Unlabeled example has {features, ?}: (x, ?)
    - Used for making predictions on new data



housingMedianAge (feature)	totalRooms (feature)	totalBedrooms (feature)
42	1686	361
34	1226	180
33	1077	271

  - Once we've trained our model with labeled examples, we use that model to predict the label on unlabeled examples. In the spam detector, unlabeled examples are new emails that humans haven't yet labeled.

What is (Supervised) ML?

- Terminology: Models
  - Model maps examples to predicted labels: y'
  - Defined by internal parameters, which are learned
  - For example, a spam detection model might associate certain features strongly with "spam".
  - Two phases of a model's life:
    - **Training** means creating or **learning** the model. That is, you show the model labeled examples and enable the model to gradually learn the relationships between features and label.
    - **Inference** means applying the trained model to unlabeled examples. That is, you use the trained model to make useful predictions (y'). For example, during inference, you can predict medianHouseValue for new unlabeled examples.

What is (Supervised) ML?

- Terminology: Regression vs. classification
  - A **regression** model predicts continuous values. For example, regression models make predictions that answer questions like the following:
    - What is the value of a house in California?
    - What is the probability that a user will click on this ad?
  - A **classification** model predicts discrete values. For example, classification models make predictions that answer questions like the following:
    - Is a given email message spam or not spam?
    - Is this an image of a dog, a cat, or a hamster?

Quiz

- Suppose you want to develop a supervised machine learning model to predict whether a given email is "spam" or "not spam." Which of the following statements are true?
  1. We'll use unlabeled examples to train the model.
  2. Words in the subject header will make good labels.
  3. The labels applied to some examples might be unreliable.
  4. Emails not marked as "spam" or "not spam" are unlabeled examples.

Quiz

- Suppose you want to develop a supervised machine learning model to predict whether a given email is "spam" or "not spam." Which of the following statements are true?
  1. We'll use unlabeled examples to train the model.
  2. Words in the subject header will make good labels.
  3. The labels applied to some examples might be unreliable.
  4. Emails not marked as "spam" or "not spam" are unlabeled examples.

Quiz

- Suppose an online shoe store wants to create a supervised ML model that will provide personalized shoe recommendations to users. That is, the model will recommend certain pairs of shoes to Marty and different pairs of shoes to Janet. Which of the following statements are true?
    1. The shoes that a user adores is a useful label.
    2. Shoe size is a useful feature.
    3. User clicks on a shoe's description is a useful label.
    4. Shoe beauty is a useful feature.

Quiz

- Suppose an online shoe store wants to create a supervised ML model that will provide personalized shoe recommendations to users. That is, the model will recommend certain pairs of shoes to Marty and different pairs of shoes to Janet. Which of the following statements are true?
  1. The shoes that a user adores is a useful label.
  2. Shoe size is a useful feature.
  3. User clicks on a shoe's description is a useful label.
  4. Shoe beauty is a useful feature.

Linear Regression

- **Linear regression** is a method for finding the straight line or hyperplane that best fits a set of points. Let’s explore linear regression intuitively before laying the groundwork for a machine learning approach to linear regression.
- It has long been known that crickets (an insect species) chirp more frequently on hotter days than on cooler days.
- For decades, professional and amateur scientists have cataloged data on chirps-per-minute and temperature.
- As a birthday gift, your Aunt Ruth gives you her cricket database and asks you to learn a model to predict this relationship. Using this data, you want to explore this relationship.

Linear Regression

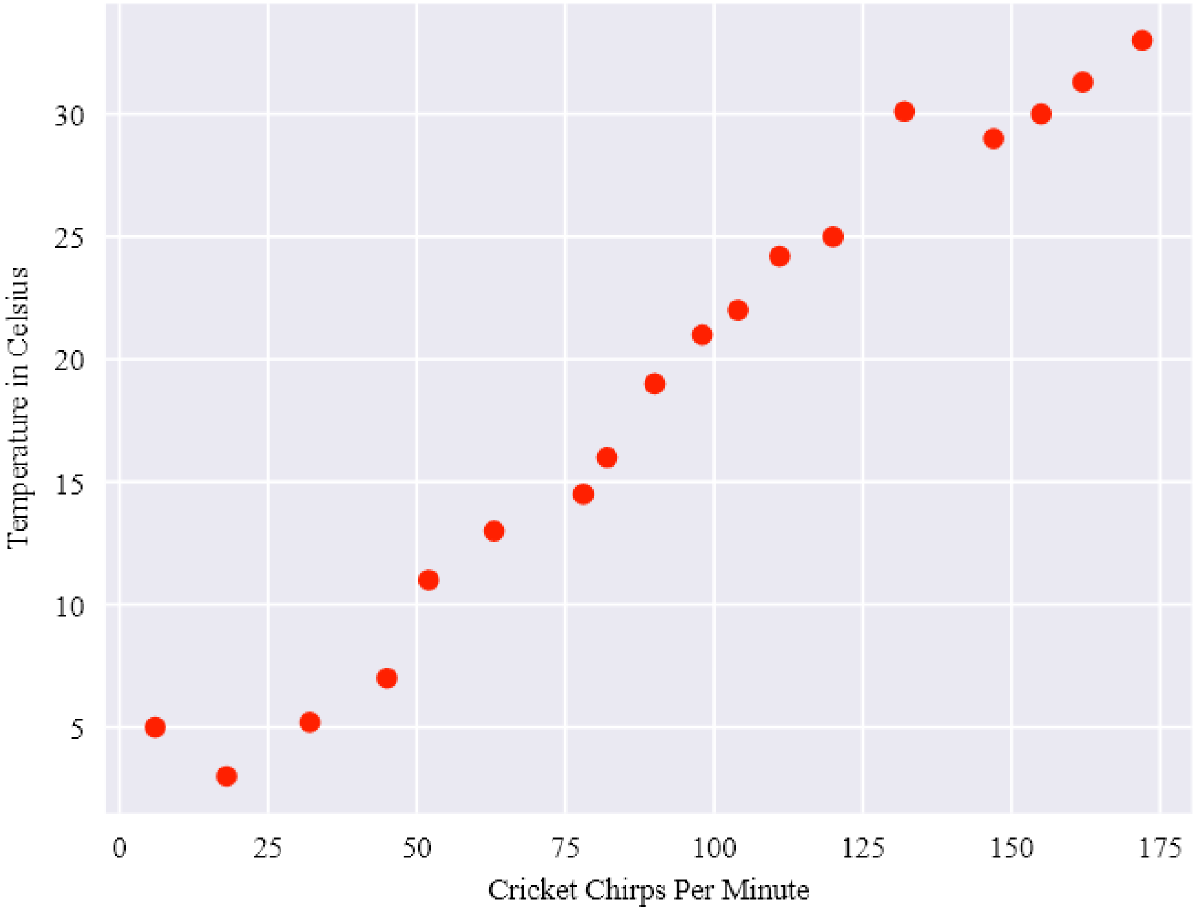


Figure 1. Chirps per Minute vs. Temperature in Celsius.

Is this relationship between chirps and temperature linear?

Linear Regression

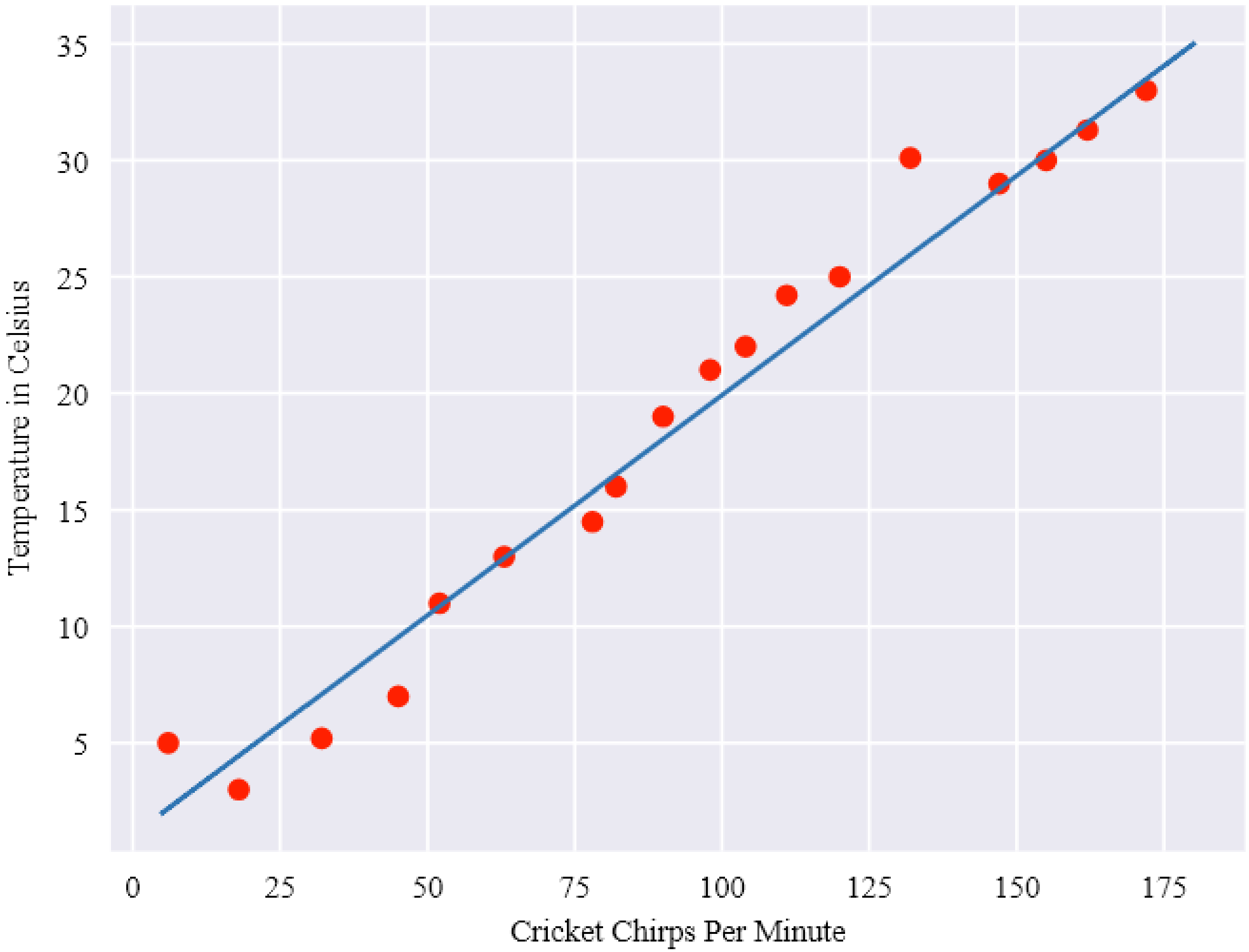


Figure 2. A linear relationship.

Linear Regression

- Using the equation for a line, you could write down this relationship as follows:

$$y=mx+b$$

where:

  - y is the temperature in Celsius—the value we're trying to predict.
  - m is the slope of the line.
  - x is the number of chirps per minute—the value of our input feature.
  - b is the y-intercept.

Linear Regression

- By convention in machine learning, you'll write the equation for a model slightly differently:

$$y' = b + w_1 x_1$$

where:

  - y′ is the predicted label (a desired output).
  - b is the bias (the y-intercept), sometimes referred to as w_0.
  - w_1 is the weight of feature 1. Weight is the same concept as the "slope" m in the traditional equation of a line.
  - x_1 is a feature (a known input).

Linear Regression

- To **infer** (predict) the temperature y’ for a new chirps-per-minute value x_1, just substitute the x_1 value into this model.
- Although this model uses only one feature, a more sophisticated model might rely on multiple features, each having a separate weight (w_1, w_2, etc.). For example, a model that relies on three features might look as follows:

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

Training and Loss

- Training:
  - **Training** a model simply means learning (determining) good values for all the weights and the bias from labeled examples. In supervised learning, a machine learning algorithm builds a model by examining many examples and attempting to find a model that minimizes loss; this process is called **empirical risk minimization**.

Training and Loss

- Loss:
  - Loss is the penalty for a bad prediction. That is, **loss** is a number indicating how bad the model's prediction was on a single example. If the model's prediction is perfect, the loss is zero; otherwise, the loss is greater. The goal of training a model is to find a set of weights and biases that have low loss, on average, across all examples.

Training and Loss

- The red arrow represents loss.
- The blue line represents predictions.

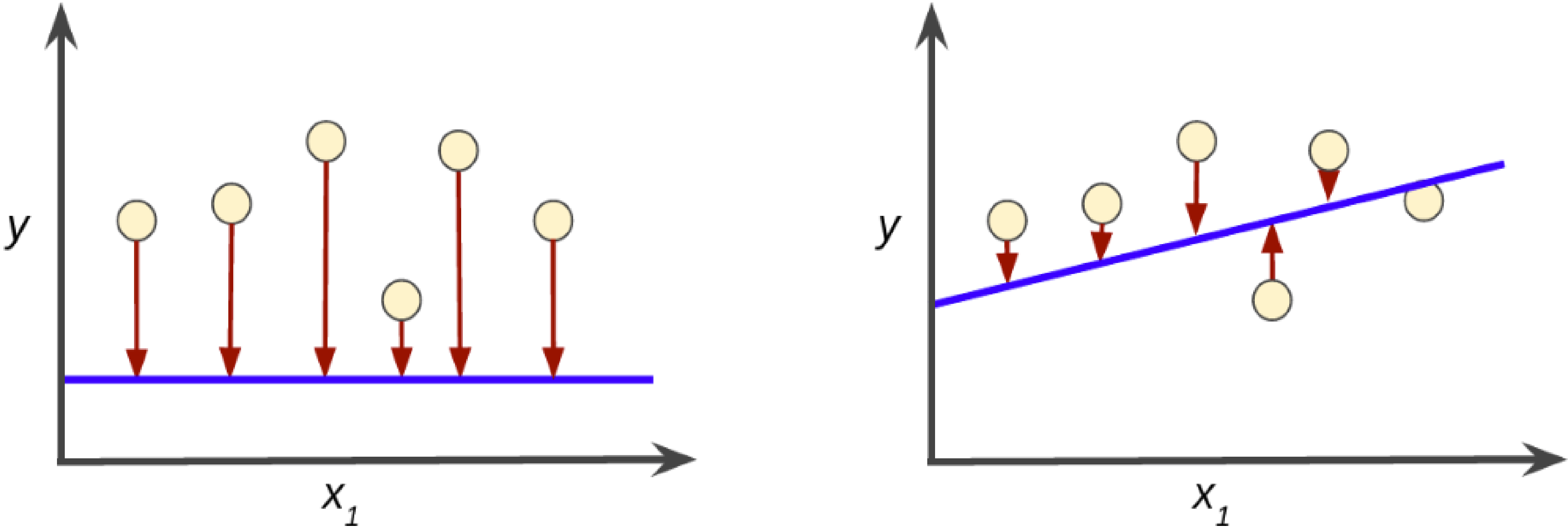


Figure 3. High loss in the left model; low loss in the right model.

- The blue line in the right plot is a much better predictive model than the blue line in the left plot.

Training and Loss

- Can you create a mathematical function—a loss function—that would aggregate the individual losses in a meaningful fashion?
- Squared loss: a popular loss function
  - The linear regression models we'll examine here use a loss function called squared loss (also known as L2 loss). The squared loss for a single example is as follows:

    = the square of the difference between the label and the prediction

    = (observation - prediction(x))2

    = (y - y')2

Training and Loss

- **Mean square error** (**MSE**) is the average squared loss per example over the whole dataset. To calculate MSE, sum up all the squared losses for individual examples and then divide by the number of examples:

$$MSE = \frac{1}{N} \sum_{(x,y)\in D} (y - prediction(x))^2$$

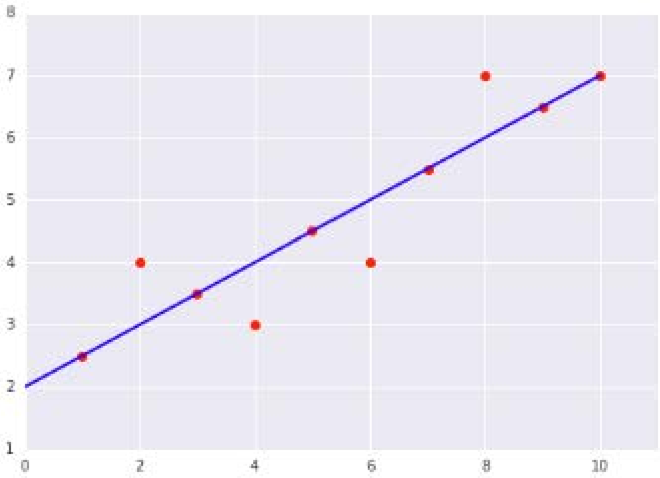
where:

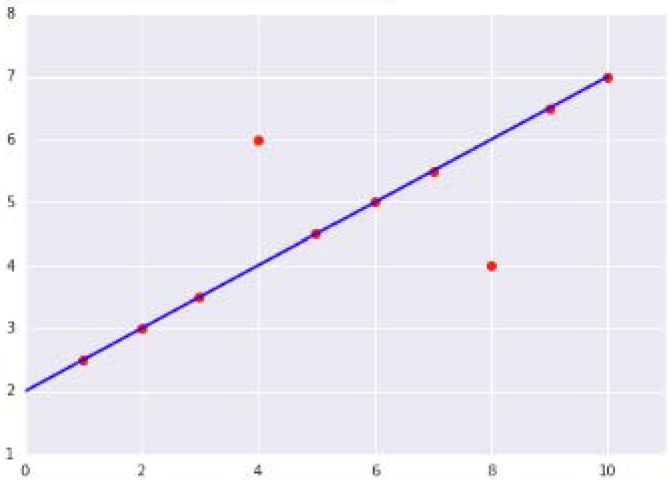
  - (x,y) is an example in which
    - x is the set of features (for example, chirps/minute, age, gender) that the model uses to make predictions.
    - y is the example's label (for example, temperature).
  - prediction(x) is a function of the weights and bias in combination with the set of features x.
  - D is a data set containing many labeled examples, which are (x,y) pairs.
  - N is the number of examples in D.

- Although MSE is commonly-used in machine learning, it is neither the only practical loss function nor the best loss function for all circumstances.

Quiz

- Consider the following two plots:

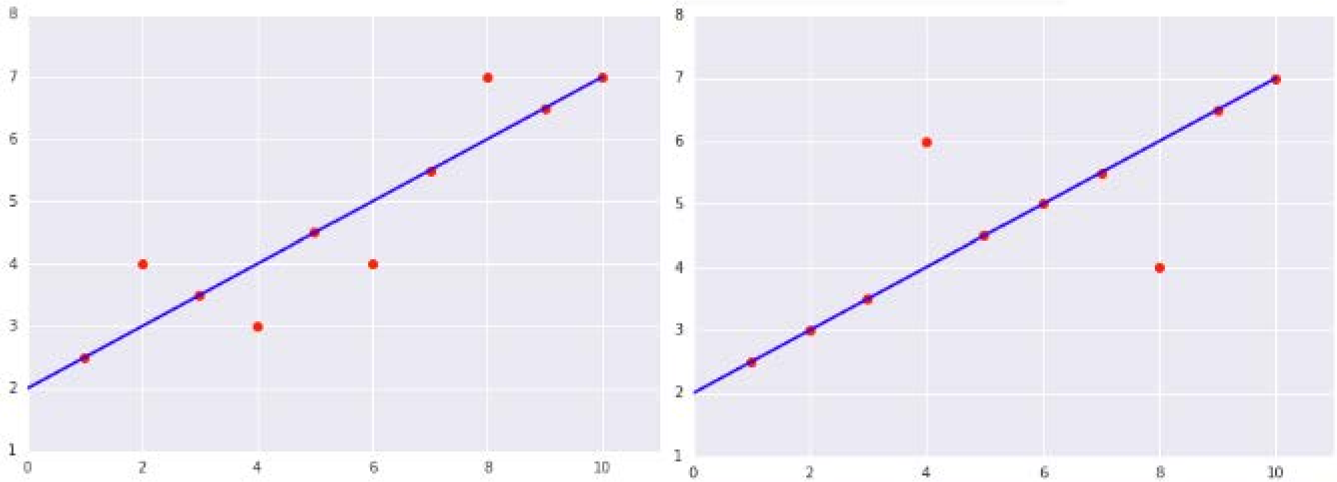




- Which of the two data sets shown in the preceding plots has the **higher** Mean Squared Error (MSE)?

Quiz

- Consider the following two plots:



- Which of the two data sets shown in the preceding plots has the **higher** Mean Squared Error (MSE)?

Left: $$MSE = \frac{0^2+1^2+0^2+1^2+0^2+1^2+0^2+1^2+0^2+0^2}{10} = 0.4$$

[✓ Right:] $$MSE = \frac{0^2+0^2+0^2+2^2+0^2+0^2+0^2+2^2+0^2+0^2}{10} = 0.8$$

Reducing Loss: An Iterative Approach

- To train a model, we need a good way to reduce the model's loss. An iterative approach is one widely used method for reducing loss, and is as easy and efficient as walking down a hill.
- The following figure suggests the iterative trial-and-error process that machine learning algorithms use to train a model:

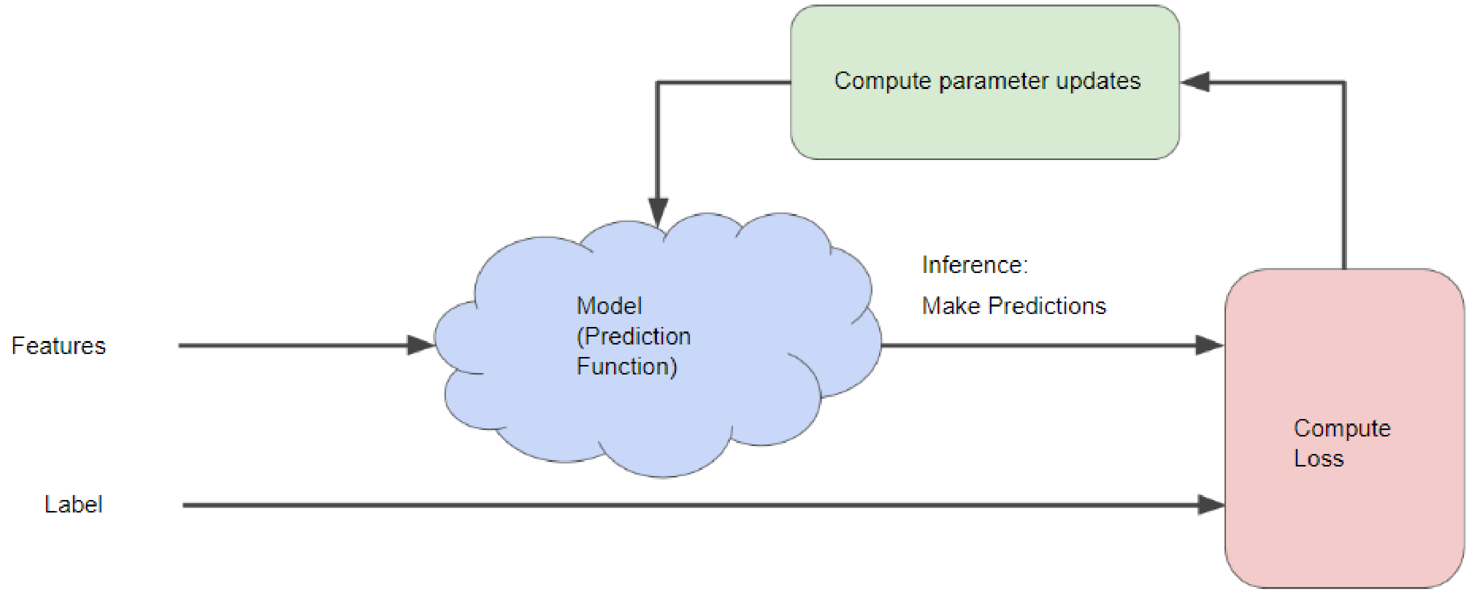


Figure 4. An iterative approach to training a model.

Reducing Loss: An Iterative Approach

- Iterative strategies are prevalent in machine learning, primarily because they scale so well to large data sets.
- The "model" takes one or more features as input and returns one prediction (y') as output.
- To simplify, consider a model that takes one feature and returns one prediction:

$$y' = b + w_1 x_1$$

- What initial values should we set for b and w_1?

Reducing Loss: An Iterative Approach

$y' = b + w_1 x_1$

- For linear regression problems, it turns out that the starting values aren't important. We could pick random values, but we'll just take the following trivial values instead:
  - $b = 0$
  - $w_1 = 0$
- Suppose that the first feature value is 10. Plugging that feature value into the prediction function yields:
  - $y' = 0 + 0(10)$
  - $y' = 0$

Reducing Loss: An Iterative Approach

- The "Compute Loss" part of the diagram is the loss function that the model will use. Suppose we use the squared loss function. The loss function takes in two input values:
  - *y'*: The model's prediction for features *x*
  - *y*: The correct label corresponding to features *x*.
- We’ve reached the "Compute parameter updates" part of the diagram.
- The machine learning system examines here the value of the loss function and generates new values for b and w_1.

Reducing Loss: An Iterative Approach

- The "Compute Loss" part of the diagram is the loss function that the model will use. Suppose we use the squared loss function. The loss function takes in two input values:
  - *y'*: The model's prediction for features *x*
  - *y*: The correct label corresponding to features *x*.
- We’ve reached the "Compute parameter updates" part of the diagram.
- The machine learning system examines here the value of the loss function and generates new values for b and w_1.

Reducing Loss: An Iterative Approach

- The machine learning system devises new values and then re-evaluates all those features against all those labels, yielding a new value for the loss function, which yields new parameter values.
- The learning continues iterating until the algorithm discovers the model parameters with the lowest possible loss.
- Usually, you iterate until overall loss stops changing or at least changes extremely slowly. When that happens, we say that the model has **converged**.

Reference

- This lecture note has been developed based on the machine learning crash course at Google, which is under *Creative Commons Attribution 3.0 License*.