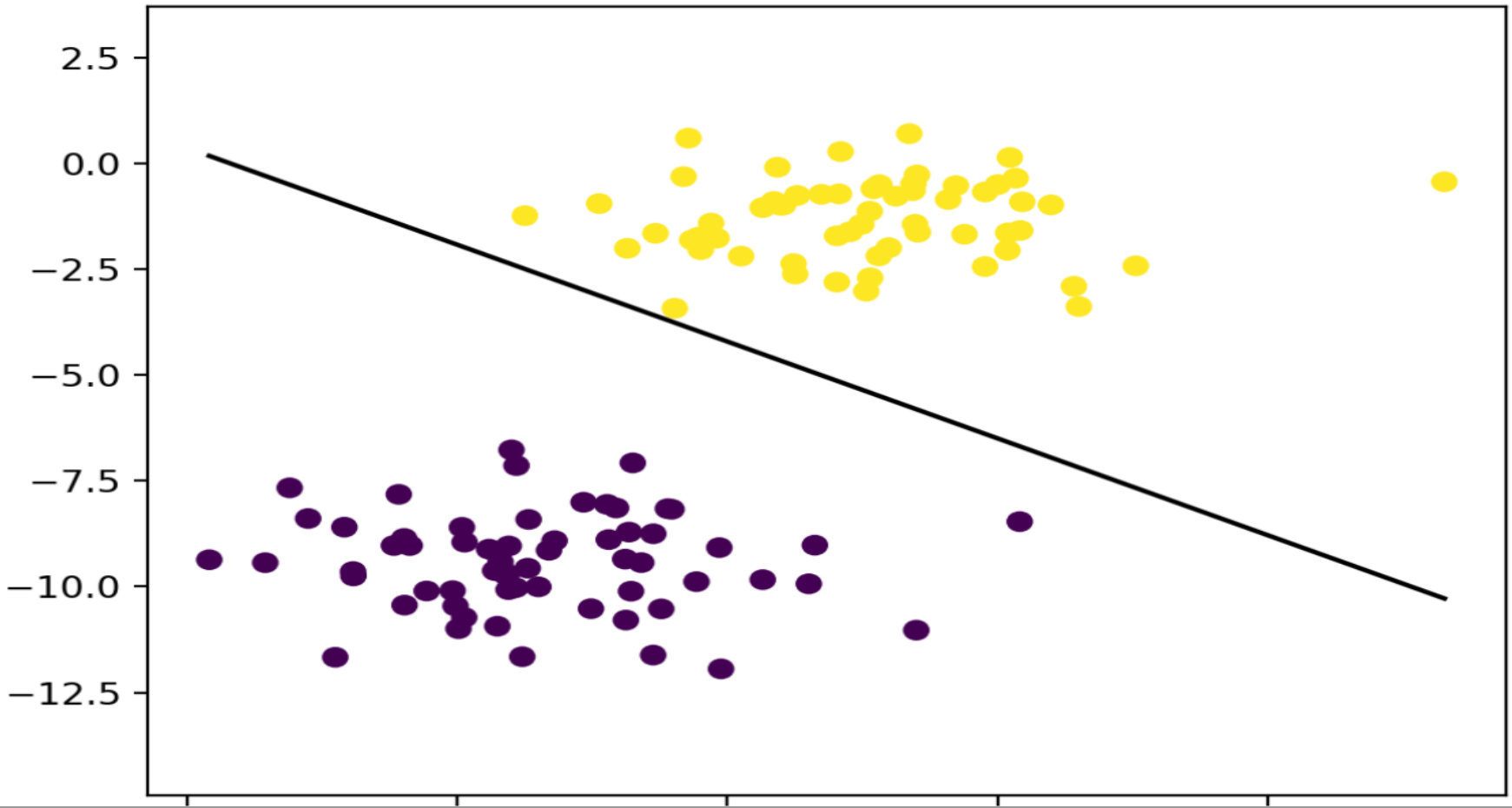
Linear Regression

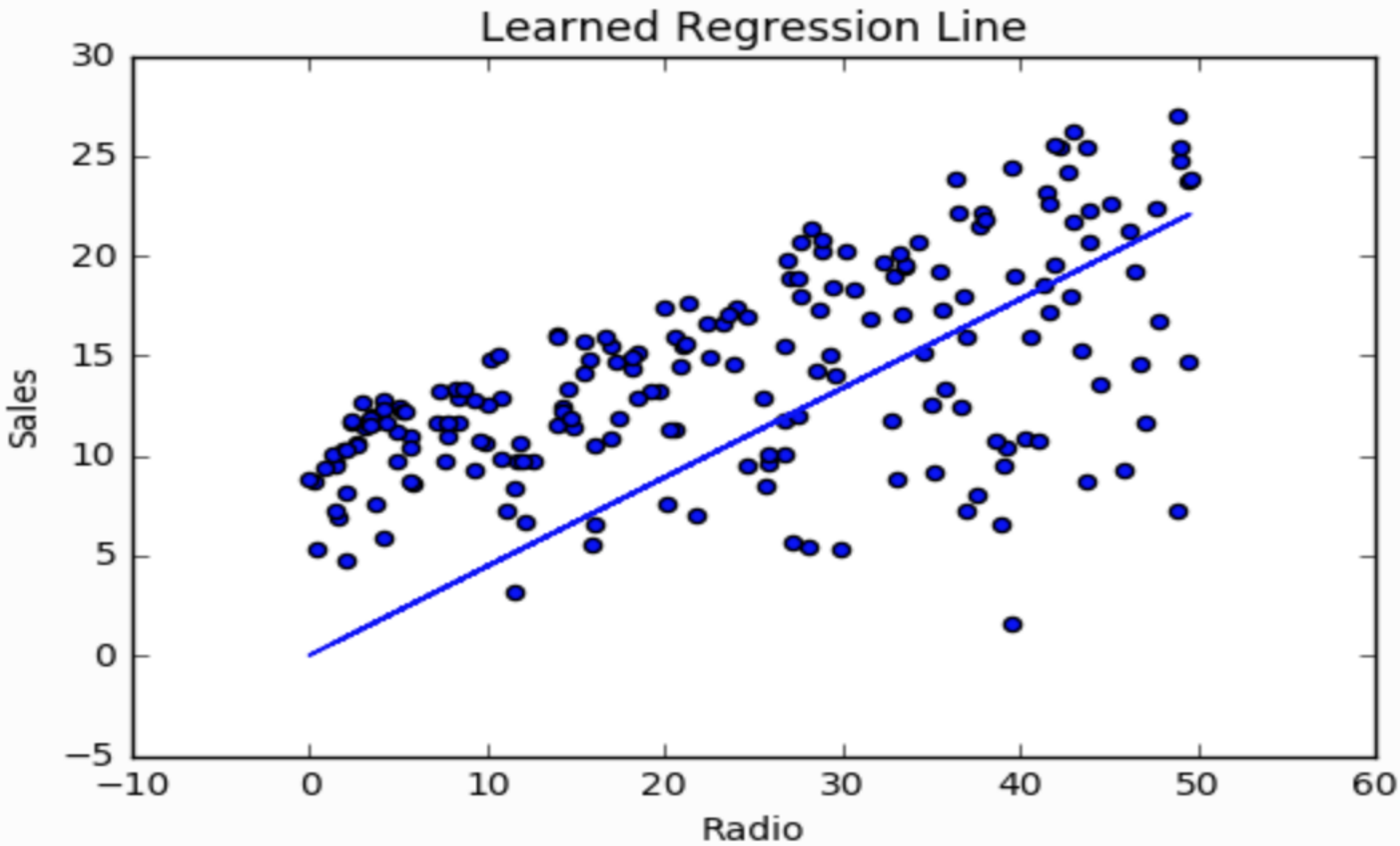
Erhard Hinrichs

Seminar für Sprachwissenschaft
Eberhard-Karls Universität Tübingen

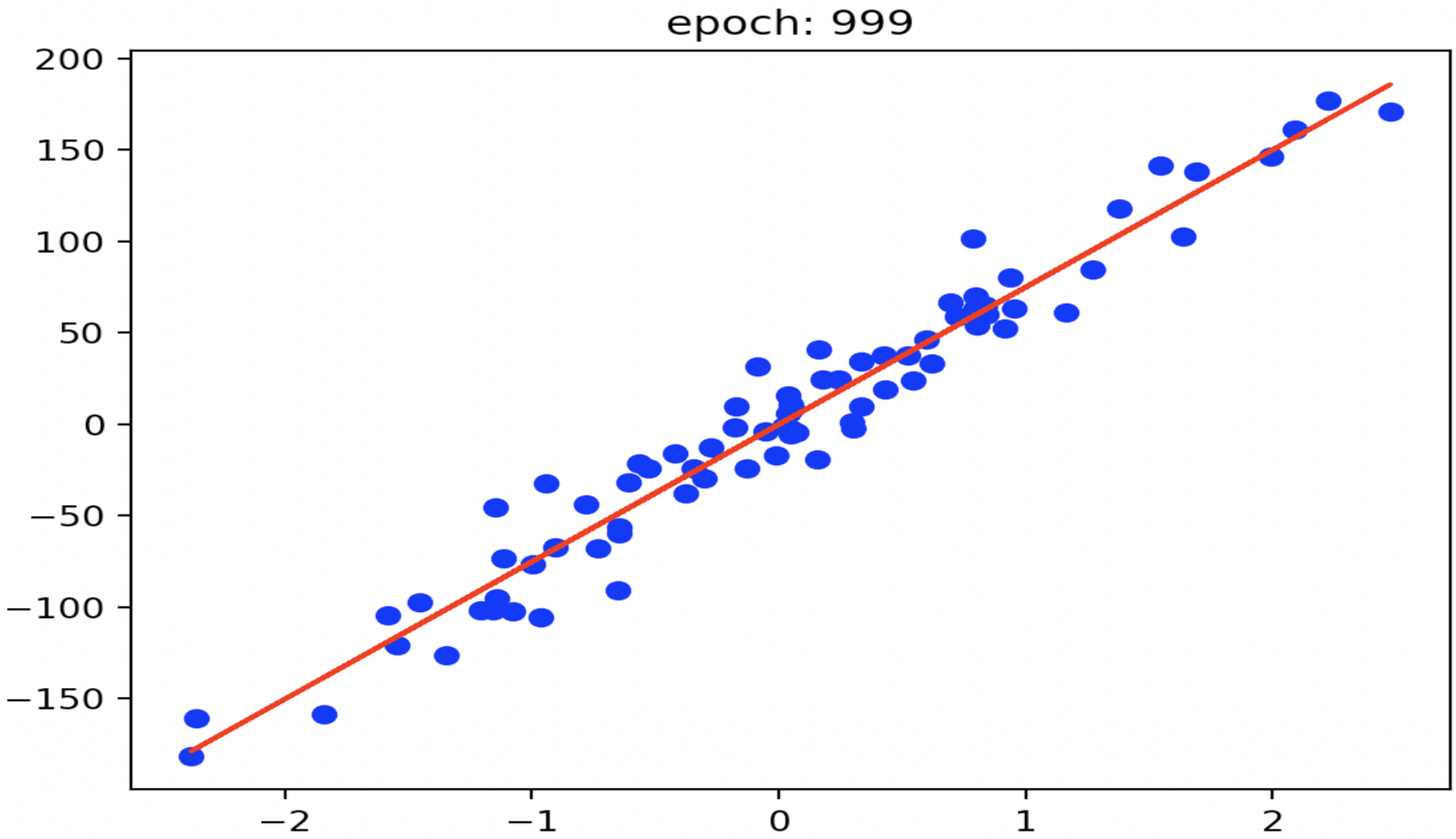
Classification by Perceptron: Example Illustration



Regression Line: Example Illustration



Regression Line: Example Illustration



Perceptron versus Linear Regression

	Classification	Regression
Type of ML model:	supervised learning	supervised learning
Output:	a class label	a numerical value
Type of Output:	categorical	continuous
Loss:	One-Zero Loss	Mean Squared Error (MSE)
	non-differentiable	differentiable function
	step function	

Some Publically Available Datasets Suitable for Linear Regression

Life Expectancy Data (WHO): sorted by country with life-expectancy data, infant-mortality rate, adult-mortality rate, per-capita expenditure on health care, average per-capita alcohol consumption, immunization coverage for hepatitis, measles infection rate, etc.

Vehicle Dataset from CarDekho for price prediction: model, year, selling price, showroom price, kilometers driven, fuel type, seller type, transmission and number of previous owners.

Real Estate Price Prediction: date of purchase, house age, location, distance to nearest MRT station, and house price of unit area.

A Linguistic Dataset Suitable for Linear Regression

Ratings Dataset (Baayen 2008): ratings of the weight, size and familiarity of a word's referent on the basis of the word's frequency, frequencies of singular and plural forms, length, class (animal or plant), famlly size, derivational entropy, synset count, complex word constituent count.

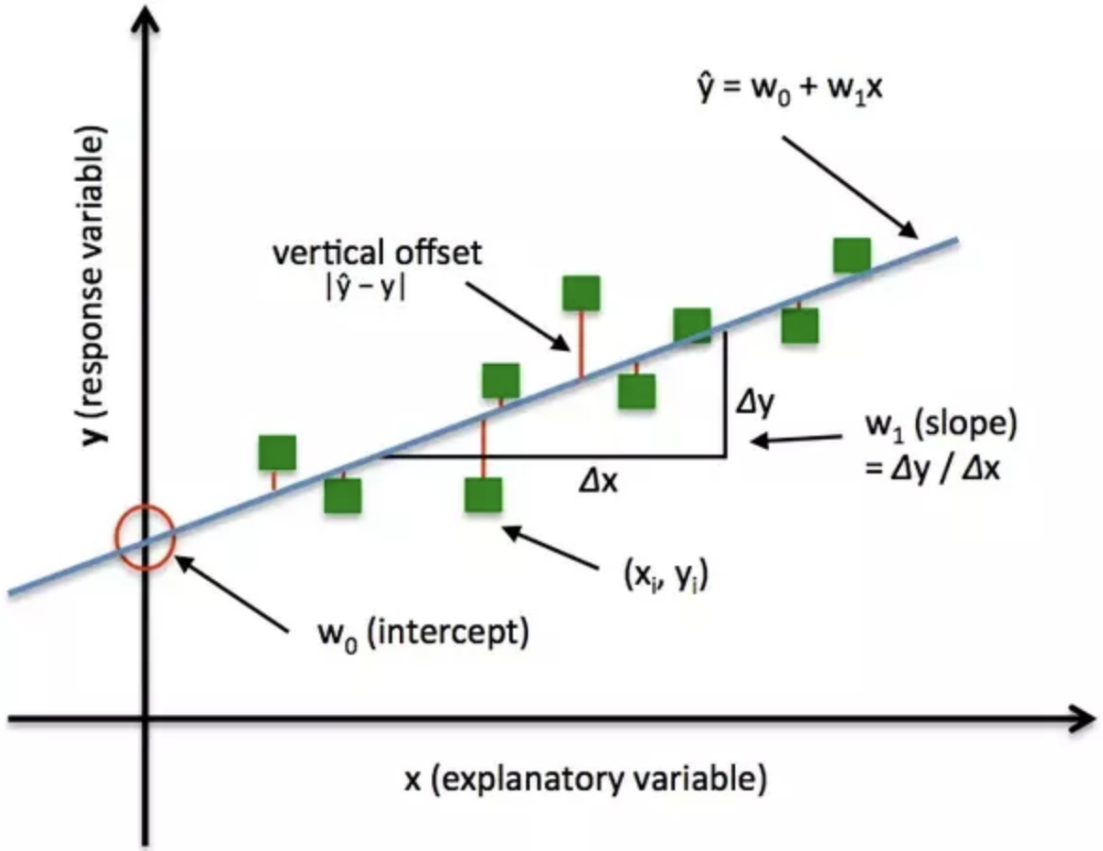
Simple versus Multivariable Regression

- Simple: $y = wx + b$
- Multivariable: $f(x, y, z) = w_1 x + w_2 y + w_3 z$

Mean squared error (MSE)

- MSE measures the average squared difference between an observation's actual and predicted values. The output is a single number representing the cost, or score, associated with our current set of weights.
- The goal is to minimize MSE to improve the accuracy of our model.
- $MSE = \frac{1}{N}\sum_{i=1}^{n}(y_i - (mx_i + b)^2$

Linear Decision Boundary: Bias and Slope



Credit:
https://algorithmia.com/blog/introduction-to-loss-functions

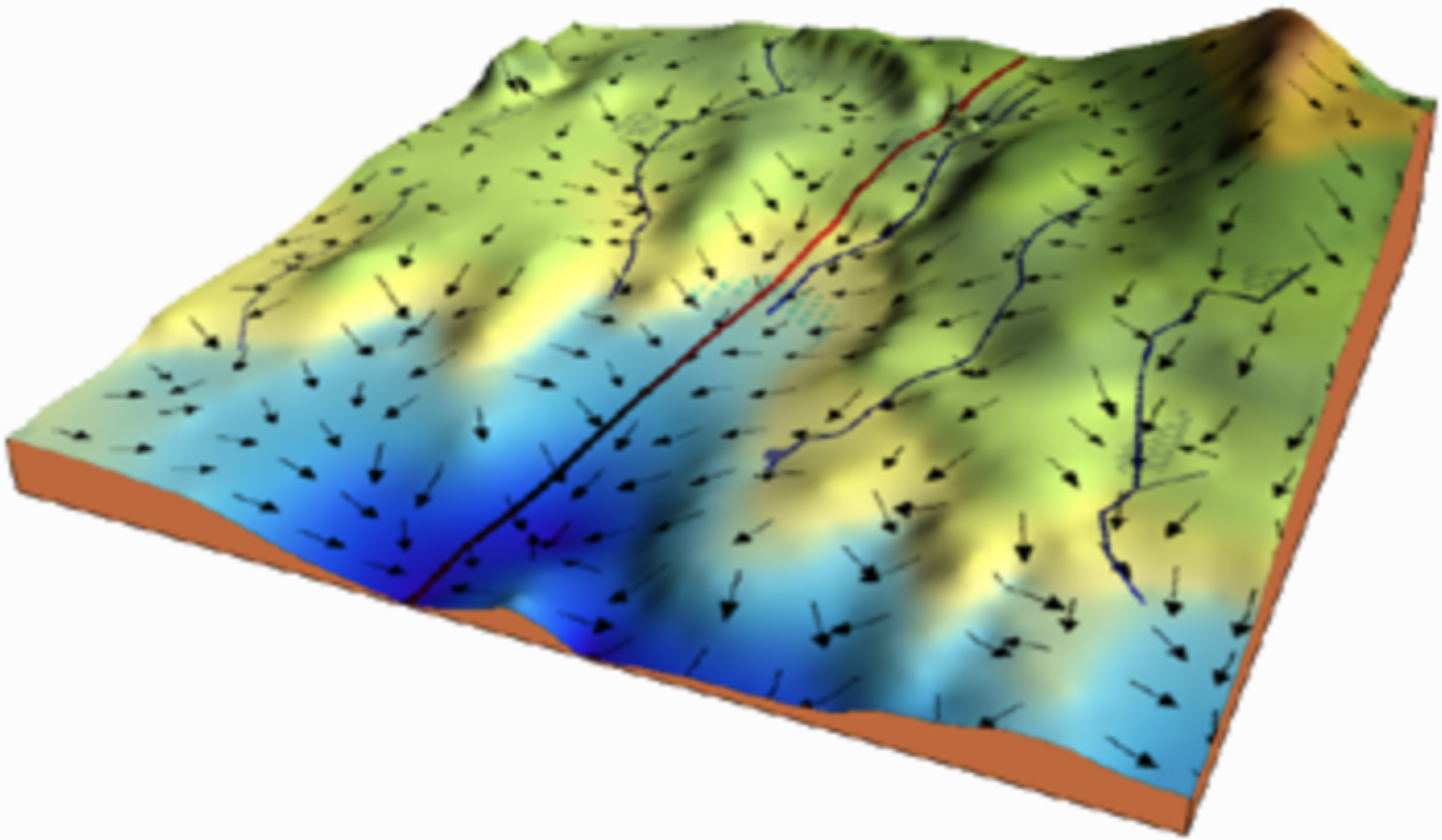
Mean Squared Error: code

```
summation = 0                          #variable to store the summation
                                       #of differences
n = len(y)                             #finding total number of items in list
for i in range (0,n):                  #looping through each element of the list
difference = y[i]                      #finding the difference between
- y_bar[i]                             # observed and predicted value
squared_difference =
   difference**2                       #taking square of the difference
summation = summation
+ squared_difference                   #taking a sum of all the differences
MSE = summation/n                      #dividing summation by total values
                                       #to obtain average
print "The MSE is: " , MSE
```

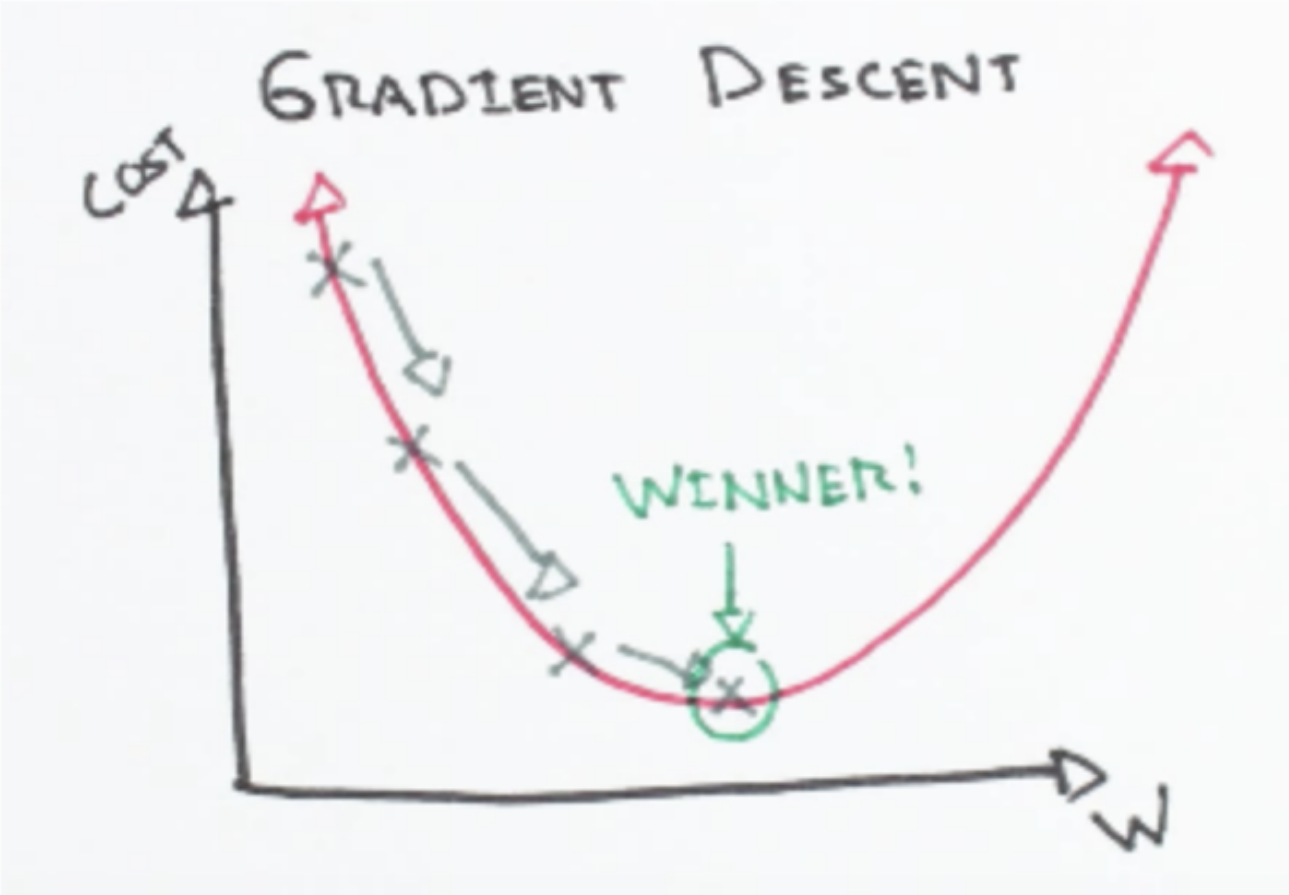
Gradient Descent

- iterative optimization algorithm for adjusting the feature weights
- uses the (partial) derivative of the cost function to find the gradient (The slope of the cost function using our current weight), and then changing our weight to move in the direction opposite of the gradient.

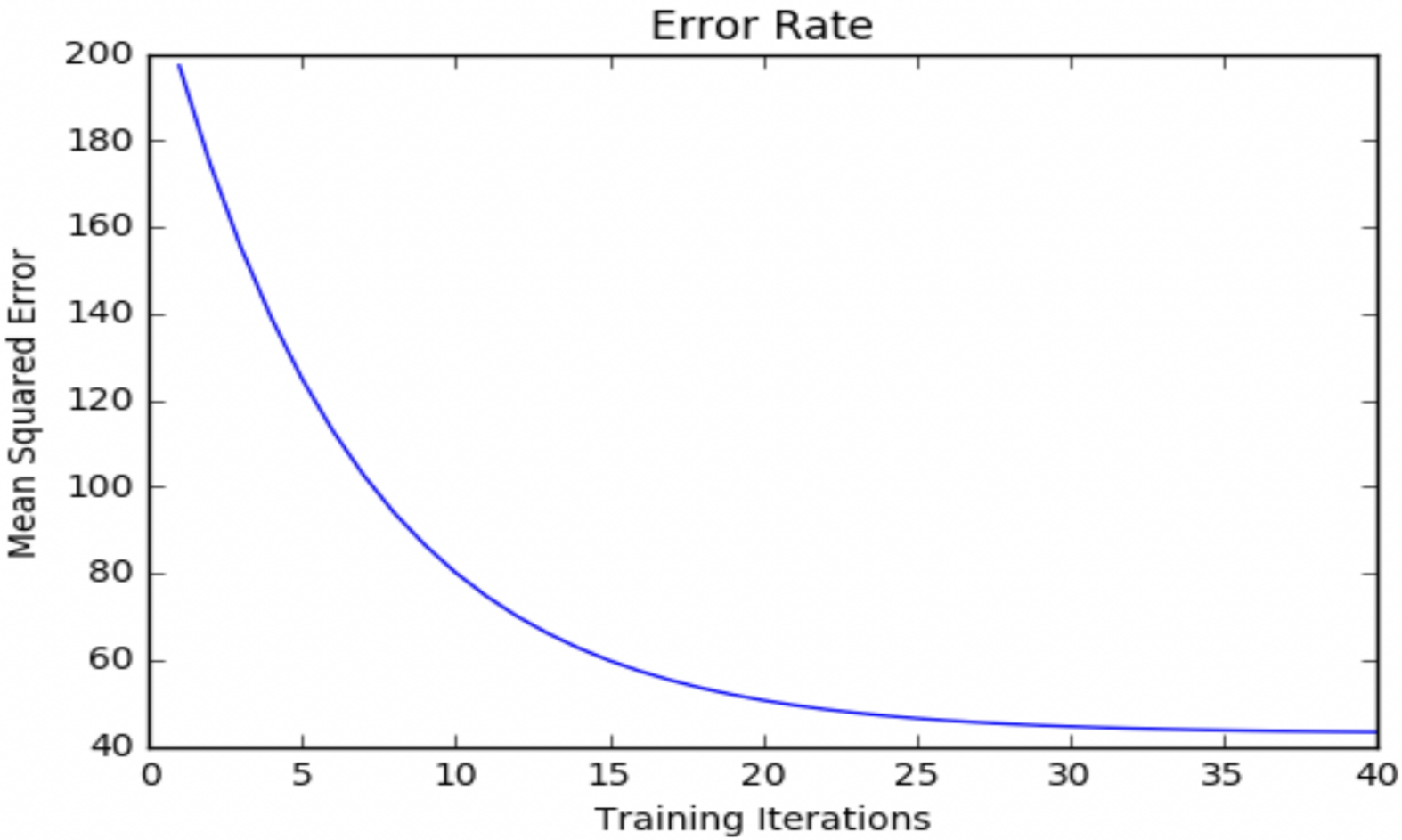
Gradient Descent: Example Illustration



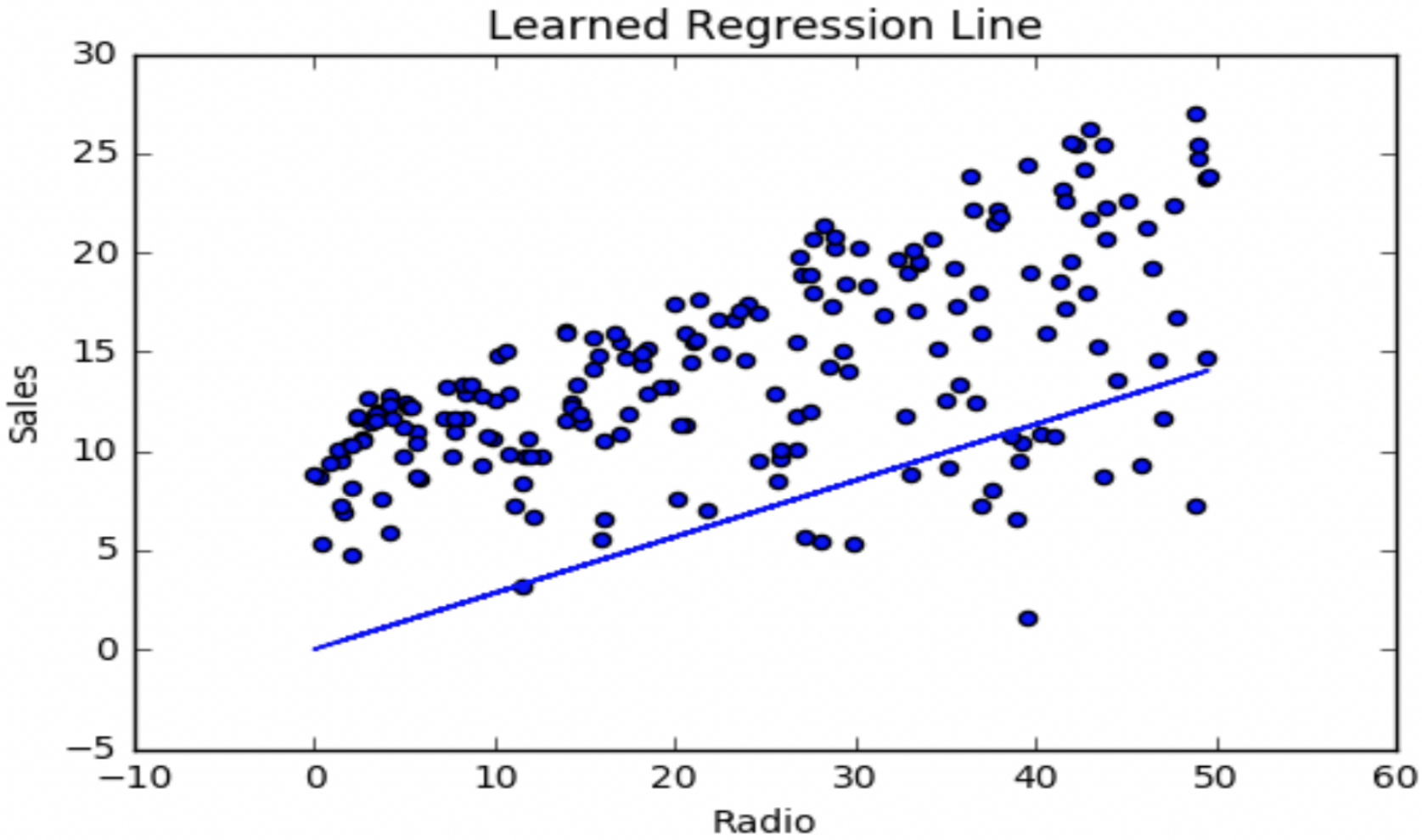
Gradient Descent: Example Illustration



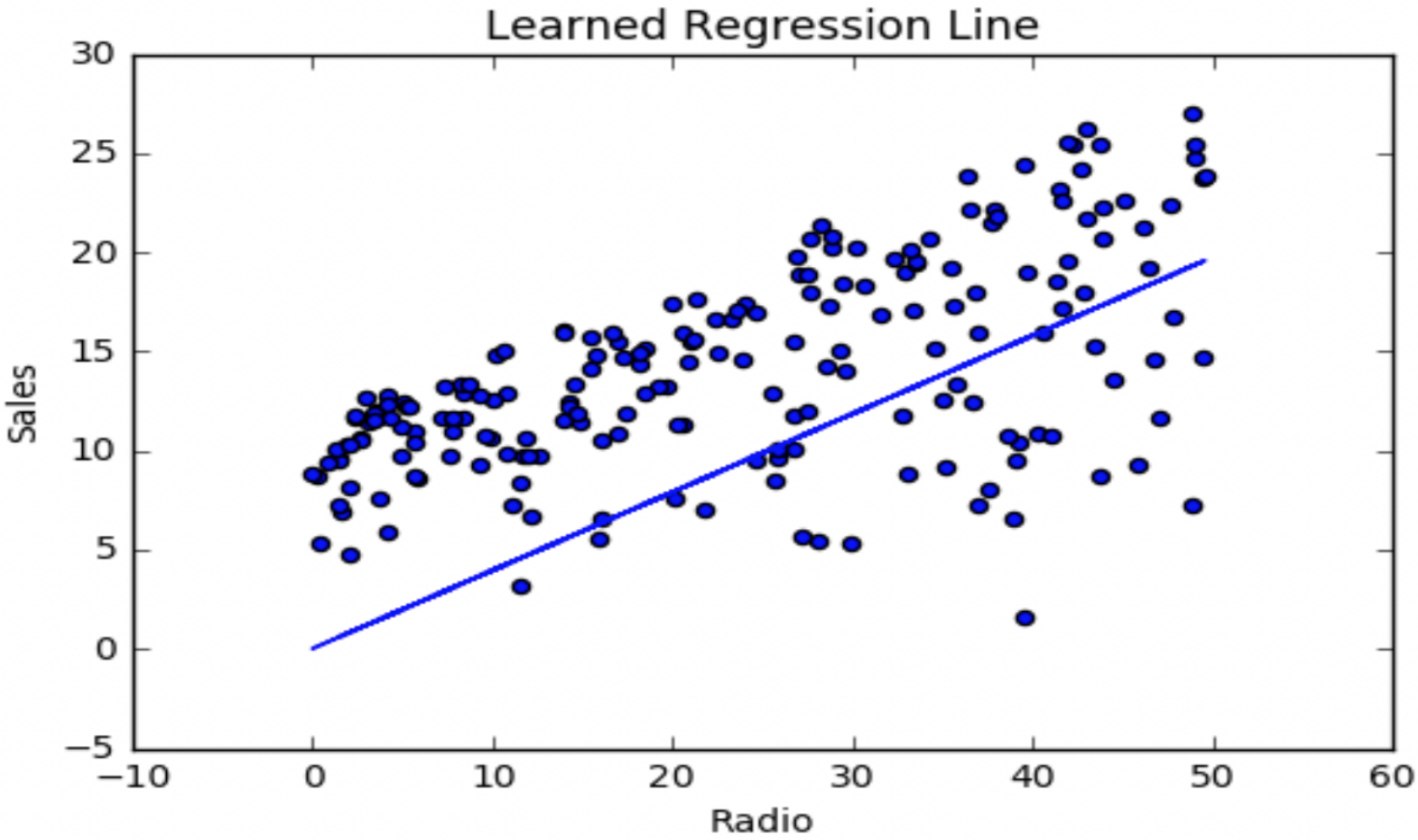
Cost History



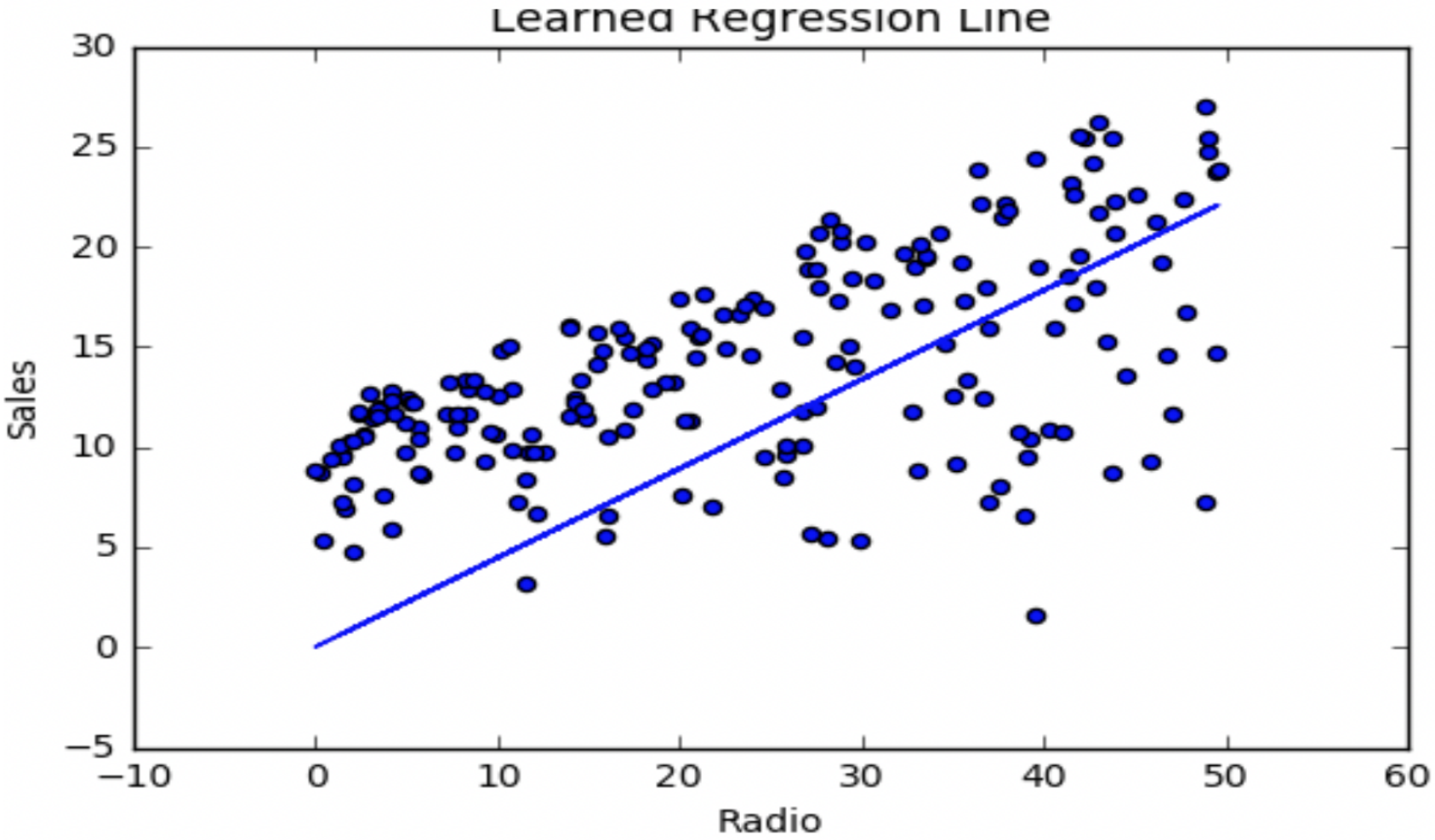
Minimizing the Cost Function



Minimizing the Cost Function



Minimizing the Cost Function



Gradient Descent: The Math

Given the cost function:

$$f(m, b) = \frac{1}{N} \sum_{i=1}^{n} (y_i - (mx_i + b))^2 \tag{1}$$

The gradient can be calculated as follows:

$$f'(m, b) = \begin{bmatrix} \frac{df}{dm} \\ \frac{df}{db} \end{bmatrix} = \begin{bmatrix} \frac{1}{N} \sum_{i=1}^{n} -2x_i(y_i - (mx_i + b)) \\ \frac{1}{N} \sum_{i=1}^{n} -2(y_i - (mx_i + b)) \end{bmatrix} \tag{2}$$

Gradient Descent: The Math of Partial Differential Equations (1)

Recall the cost function:

$$MSE = f(m, b) = \frac{1}{N} \sum_{i=1}^{n} (y_i - (mx_i + b))^2$$

There are two parameters (coefficients) in the cost function that we want to optimize via gradient descent: weight **m** and bias **b**, using partial derivatives.
To find the partial derivatives, the **chain rule** is utilized. The chain rule is needed because the formula $(y - (mx + b))^2$ consists two nested functions: the inner function $(y - (mx + b)$ and the outer function x^2.

The Chain Rule

$$\frac{df}{dx} = \frac{df}{du} \cdot \frac{du}{dx}$$

In-class Example and Exercise:

$$f(x) = (x^2)^3$$

Question: Given $f(x) = (x^2)^3$, what is $f'(x)$?

Is it: $f'(x) = 3x^2$?

Or is it $f'(x) = 6x^5$?

Or is it $f'(x) = 5x^4$?

Or is it: none of the above?

The Chain Rule: Example

Question: Given $f(x) = (x^2)^3$, what is $f(x)'$? Answer is: $6x^5$

Step-by-step: $\frac{d}{dx} f(x) = f'((x^2)^3)$

Chain-rule: $\frac{df}{dx} = \frac{df}{du} \cdot \frac{du}{dx}$

replace x^2 by u; i.e. $u = x^2$, and set $f(x) = u^3$

$\frac{d}{du} u^3 * \frac{d}{dx} x^2 = 3u^2 \cdot 2x$

replace u by x^2: $3(x^2)^2 \cdot 2x = 3x^4 \cdot 2x = 6x^5$

The Product Rule

$$\frac{d}{dx}(u \cdot v) = \frac{du}{dx} \cdot v + u \cdot \frac{dv}{dx} \tag{3}$$

The Quotient Rule

Let $h(x) = \frac{f(x)}{g(x)}$, where f and g are differentiable and $g(x) \neq 0$.

$$h'(x) = \frac{f'(x) \cdot g(x) - f(x) \cdot g'(x)}{g(x)^2} \tag{4}$$

Derivative of Natural Logarithm

$$\frac{d}{dx}(ln\ x) = \frac{1}{x} \qquad for\ x > 0 \tag{5}$$

Linearity of Differentiation

$$\frac{d(af + bg)}{dx} = a \cdot \frac{df}{dx} + b \cdot \frac{dg}{dx} \tag{6}$$

Special Cases:

$$(a\ f)' = a \cdot f' \tag{7}$$

$$(f + g)' = f' + g' \tag{8}$$

$$(f - g)' = f' - g' \tag{9}$$

The General Power Rule

$$\frac{d}{dx}[[f(x)]^n] = n[f(x)]^{n-1} * \frac{d}{dx}[f(x)]$$

The General Power Rule: Example

$y = (x^3 + 2x + 38)^4$

$\frac{d}{dx}(x^3 + 2x + 38)^4 = 4(x^3 + 2x + 38)^3 * \frac{d}{dx}[x^3 + 2x + 38]$

$= 4(x^3 + 2x + 38)^3 * (3x^2 + 2) = (12x^2 + 8) * (x^3 + 2x + 38)^3$

Gradient Descent: The Math of Partial Differential Equations (2)

The cost function:
$f(m, b) = \frac{1}{N} \sum_{k=1}^{n} (y_i - (mx_i + b))^2$
can be re-written as:
$(y_i - (mx_i + b))^2 = \mathrm{A(B(m,b))}$

We can split the derivative into $A(x) = x^2$ and $\frac{df}{dx} = A'(x) = 2x$

and

$B(m, b) = y_i - (mx_i + b) = y_i - mx_i - b$ with:
$\frac{dx}{dm} = B'(m) = 0 - x_i - 0 = -x_i$
$\frac{dx}{db} = B'(b) = 0 - 0 - 1 = -1$

Gradient Descent: The Math of Partial Differential Equations (3)

Using the Chain Rule, we obtain the following:

$$\frac{df}{dm} = A'(B(m,f) * B'(m) = 2(y_i - (mx_i + b)) * -x_i$$
$$\frac{df}{db} = A'(B(m,f) * B'(b) = 2(y_i - (mx_i + b)) * -1$$

$$f'(m,b) = \begin{bmatrix} \frac{df}{dm} \\ \frac{df}{db} \end{bmatrix} = \begin{bmatrix} \frac{1}{N}\sum_{i=1}^{n} -x_i * 2(y_i - (mx_i + b)) \\ \frac{1}{N}\sum_{i=1}^{n} -1 * 2(y_i - (mx_i + b)) \end{bmatrix} \tag{10}$$

$$f'(m,b) = \begin{bmatrix} \frac{df}{dm} \\ \frac{df}{db} \end{bmatrix} = \begin{bmatrix} \frac{1}{N}\sum_{i=1}^{n} -2x_i(y_i - (mx_i + b)) \\ \frac{1}{N}\sum_{i=1}^{n} -2(y_i - (mx_i + b)) \end{bmatrix} \tag{11}$$

Gradient Descent for Simple Regression

Given the cost function:

$$f(m, b) = \frac{1}{N} \sum_{i=1}^{n} (y_i - (mx_i + b))^2 \tag{12}$$

The gradient can be calculated as follows:

$$f'(m, b) = \begin{bmatrix} \frac{df}{dm} \\ \frac{df}{db} \end{bmatrix} = \begin{bmatrix} \frac{1}{N} \sum_{i=1}^{n} -2x_i(y_i - (mx_i + b)) \\ \frac{1}{N} \sum_{i=1}^{n} -2(y_i - (mx_i + b)) \end{bmatrix} \tag{13}$$

Generalizing to Multi-variable Regression

For multi-variable regression, involving three variables, the cost function looks like this:

$$f(m, b) = \frac{1}{2N} \sum_{i=1}^{n} (y_i - (W_1 x_1 + W_2 x_2 + W_3 x_3))^2$$

Using the chain rule, we can compute the gradient as a vector of partial derivatives for each weight as follows:

$$f'(W_1) = -x_1(y - (W_1)x_1 + (W_2)x_2 + (W_3)x_3))$$

$$f'(W_2) = -x_2(y - (W_1)x_1 + (W_2)x_2 + (W_3)x_3))$$

$$f'(W_3) = -x_3(y - (W_1)x_1 + (W_2)x_2 + (W_3)x_3))$$

Generalizing to Multi-variable Regression

$$f'(W_1) = -x_1(y - (W_1)x_1 + (W_2)x_2 + (W_3)x_3))$$
$$f'(W_2) = -x_2(y - (W_1)x_1 + (W_2)x_2 + (W_3)x_3))$$
$$f'(W_3) = -x_3(y - (W_1)x_1 + (W_2)x_2 + (W_3)x_3))$$

resulting in the following vector of gradients for weight updating:

$$f'(m) = \begin{bmatrix} \frac{df}{dW_1} \\ \frac{df}{dW_2} \\ \frac{df}{dW_3} \end{bmatrix} = \begin{bmatrix} \frac{1}{2N}\sum_{i=1}^{n} -x_1(y - (W_1)x_1 + (W_2)x_2 + (W_3)x_3)) \\ \frac{1}{2N}\sum_{i=1}^{n} -x_2(y - (W_1)x_1 + (W_2)x_2 + (W_3)x_3)) \\ \frac{1}{2N}\sum_{i=1}^{n} -x_3(y - (W_1)x_1 + (W_2)x_2 + (W_3)x_3)) \end{bmatrix} \quad (14)$$