Linear Regression

Christian Darvin

June 3, 2025

Contents

1 Linear Regression

A statistical technique used to find the relationship between **features** and a **label**. Possible correlations: **Positive, Negative, Non-linear, No Correlation**.

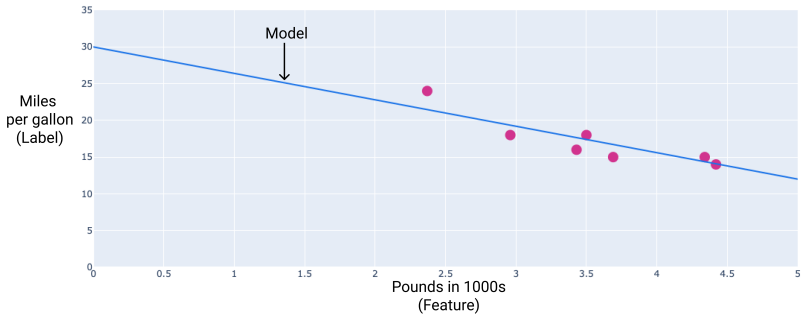


Figure 1: Linear Regression Plot

1.1 Linear Regression Equation

$$\hat{y} = b + w_1 x_1 \tag{1}$$

where:

$\hat{y}$: predicted label (output)
b: bias/parameter is same as the y-intercept in algebra (shifts the line vertically)
w_1: weight/parameter is same as the slope m in algebra (steepness of the line)
x_1: feature (input)

During training, the model calculates the w_1 and b that produce the best model.

1.2 Models with Multiple Features

$$\hat{y} = b + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n \tag{2}$$

$x_1, \ldots, x_5 = \{$Pounds, Displacement, Acceleration, Number of Cylinders, Horsepower$\}$

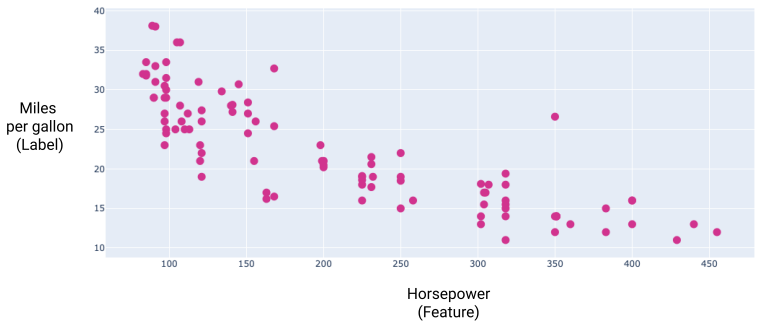


Figure 2: Horsepower Plot

2 Loss Function

A metric that describes how wrong a model's predictions are. It measures the distance between the model's predictions and the actual values. The objective is to minimize the loss, making it to its lowest possible value.

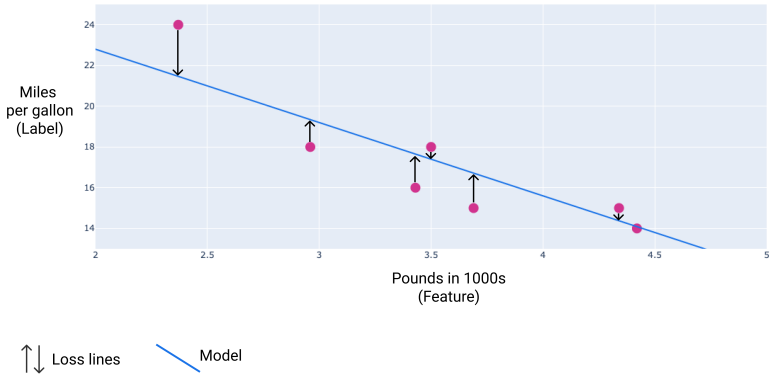


Figure 3: Loss Lines

2.1 Distance of Loss

Loss focuses on the distance, not the value. If the difference is a negative value, we need to remove the sign.
Common methods to remove the sign:

- Get the absolute value of the difference of errors
- Square the difference of errors

2.2 Error

The difference between the actual and predicted values

$$error = y - \hat{y} \tag{3}$$

2.3 Types of Loss

Loss Type	Definition	Equation
L_1 loss	Sum of the absolute values of the errors	$\sum \lvert y - \hat{y} \rvert$
L_2 loss	Sum of the squared difference of the errors	$\sum (y - \hat{y})^2$
Mean Absolute Error (MAE)	Average of L_1 losses across N	$\frac{1}{N} \sum \lvert y - \hat{y} \rvert$
Mean Squared Error (MSE)	Average of L_2 losses across N	$\frac{1}{N} \sum (y - \hat{y})^2$

It is recommended to use MAE or MSE.

2.4 Loss Calculation Example

Calculate the L_2 loss:
w_1: -3.6
b: 30
If a model predicts that a 2,370-pound car gets 21.5 miles per gallon, but it actually gets 24 miles per gallon, then:

$$\begin{aligned} \hat{y} &= b + (w_1 + \text{feature value}) && (4)\\ &= 30 + (-3.6 * 2.37) && (5)\\ &= 21.5 && (6) \end{aligned}$$

$y = 24$, $\hat{y} = 21.5$

$$\begin{aligned} L_2 &= (y - \hat{y})^2 && (7)\\ &= (24 - 21.5)^2 && (8)\\ &= 6.25 && (9) \end{aligned}$$

2.5 Choosing a Loss

An outlier is a value that lies outside the typical range of a dataset. **Mean Squared Error (MSE)** is more sensitive to outliers, giving them greater influence on the loss. In contrast, **Mean Absolute Error (MAE)** is less affected by outliers and stays closer to the majority of the data points.

3 Gradient Descent

Gradient Descent is an iterative process used to update the weight and bias in order to minimize the loss.

1. Initialize the w and b to 0.
2. Compute the loss function using the current w and b.
3. Compute the gradients of the loss with respect to w and b, then update the parameters by moving them a small step in the direction that decreases the loss (using the learning rate).
4. Repeat steps 2 and 3 until convergence (when the loss stops decreasing significantly or after a set number of iterations).

3.1 Math Behind Gradient Descent

Pounds in 1000s = [3.5, 3.69, 3.44, 3.43, 4.34, 4.42]
Miles per gallon = [18, 15, 18, 16, 15, 14, 24]

1. Initialize the w and b to 0.

$$w = 0 \tag{10}$$
$$b = 0 \tag{11}$$
$$\hat{y} = 0 + 0(x_1) \tag{12}$$

2. Calculate MSE loss with current model parameters

$$\text{MSE} = \frac{(18-0)^2 + (15-0)^2 + (18-0)^2 + (16-0)^2 + (15-0)^2 + (14-0)^2 + (24-0)^2}{7} \tag{13}$$
$$= 303.71 \tag{14}$$

3. Calculate the slope of the tangent to the loss function at each weight and the bias

$$\text{w slope} = -119.7 \tag{15}$$
$$\text{b slope} = -34.3 \tag{16}$$

How? Derivatives...

4. Move a small amount (learning rate: 0.01) in the direction of the negative slope to get the next w and b.

$$w = w - (0.01 * -119.7) \tag{17}$$
$$b = b - (0.01 * -34.3) \tag{18}$$
$$w_{\text{new}} = 1.2 \tag{19}$$
$$b_{\text{new}} = 0.34 \tag{20}$$
$$\tag{21}$$

Iteration	**Weight**	**Bias**	**Loss (MSE)**
1	0.00	0.00	303.71
2	1.20	0.34	170.67
3	2.75	0.59	67.30
4	3.17	0.72	50.63
5	3.47	0.82	42.10
6	3.68	0.90	37.74

Table 1: Training Progress Over Iterations

Continue training until the loss has stabilized.

3.2 Loss Curves

The loss curve shows how the loss changes as the model trains.

- **x-axis**: iterations
- **y-axis**: loss

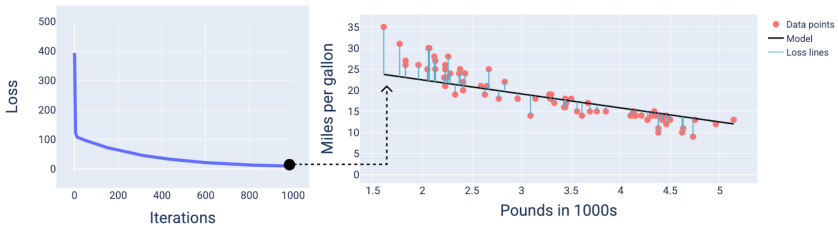


Figure 4: Low Loss Curve

3.3 Convergence and Convex Functions

The loss functions for **linear models** always produce a **convex** surface.

- **x-axis**: weight
- **y-axis**: bias
- **z-axis**: loss

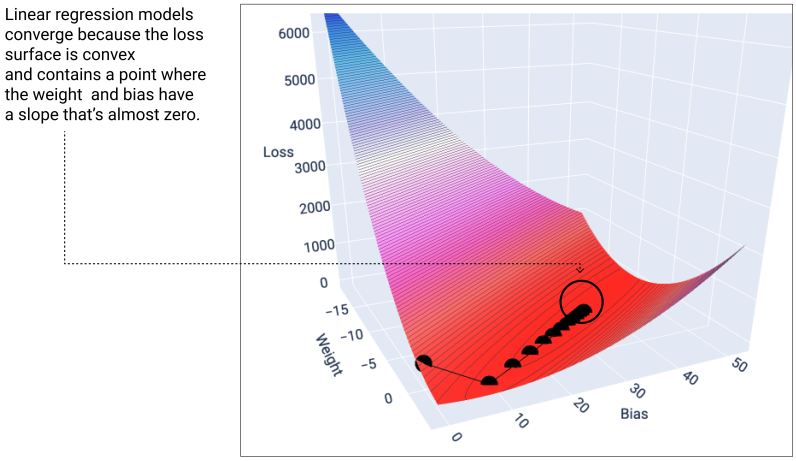


Figure 5: Loss Surface Points

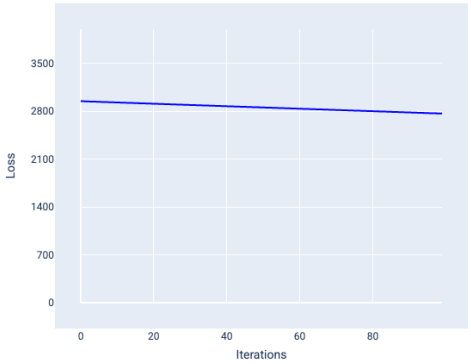
A linear model converges when it reaches the point of minimum loss. After this, further iterations cause only minor adjustments to the weights and bias. This process is like a ball rolling downhill and settling at the lowest point. While the model may not find the exact minimum, it gets very close. Importantly, this minimum doesn't mean zero loss—just the lowest possible loss for the given parameters. It will never reach zero, but close to zero. 0.00310

4 Hyperparameters

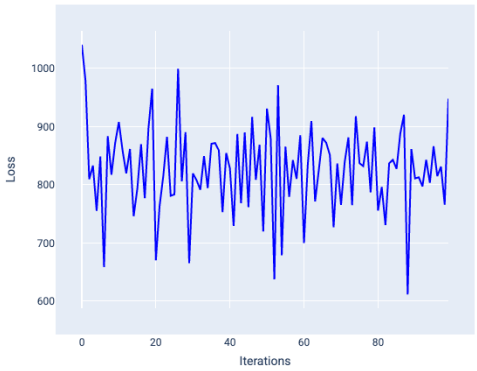
Hyperparameters are values that you control. **Parameters** are values that the model calculates during training.

4.1 Learning Rate

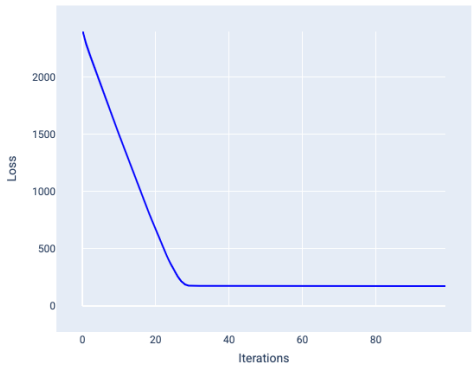
It influences how quick the model converges.
Learning rate: too low - the model will take a long time to converge.
Learning rate: too high - the model may never converge and will bounce around the values of w and b that minimize the loss.
Common learning rates: {0.1, 0.01, 0.001, 0.0001}, depending on the model and dataset.



(a) Small Learning Rate



(b) Large Learning Rate



(c) Correct Learning Rate

Figure 6: Comparison of Learning Rates

4.2 Batch Size

Number of examples the model processes before updating its w and b. When a dataset contains thousands or even millions of examples, using the full batch isn't practical.

4.2.1 Stochastic Gradient Descent (SGD)

Uses only a single example (a batch size of one) per iteration. One example comprising each batch is chosen at random. It is **Noisy**. "Noise" refers to variations during training that cause the loss to increase rather than decrease during an iteration.

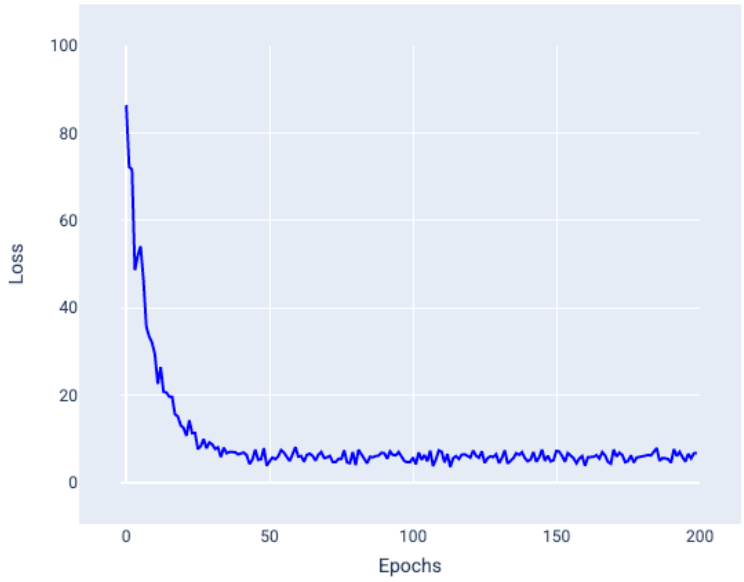


Figure 7: Stochastic Gradient Descent

4.2.2 Mini-batch Stochastic Gradient Descent (mini-batch SGD)

For N number of data points, the batch size is between 1 and $N-1$. The model chooses the examples included in each batch at random, averages their gradients, and then updates the weights and bias once per iteration.

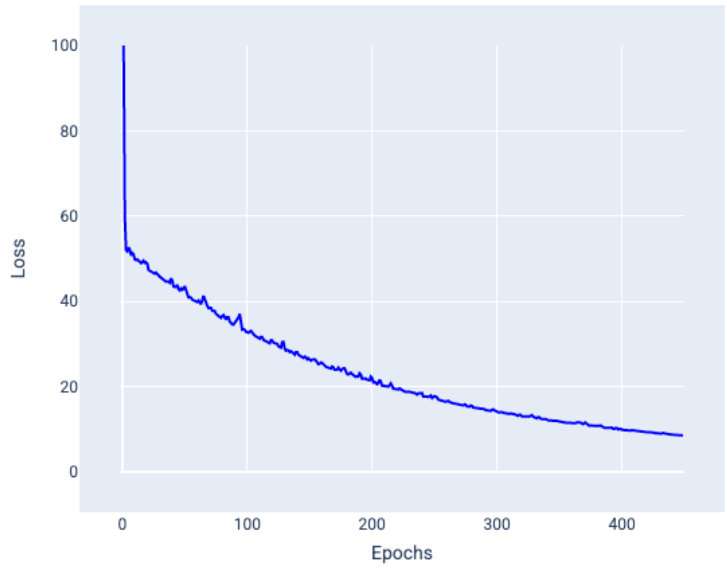


Figure 8: Mini-batch Stochastic Gradient Descent

4.3 Epochs

It means that the model has processed every example in the training set once. More epochs produces a better model, but also takes more time to train.

Batch Type	When parameters update occur
Full batch	Updates weights once per epoch after using the entire dataset
Stochastic Gradient Descent	Updates weights after each individual example
Mini-batch SGD	Updates weights after each mini-batch of examples

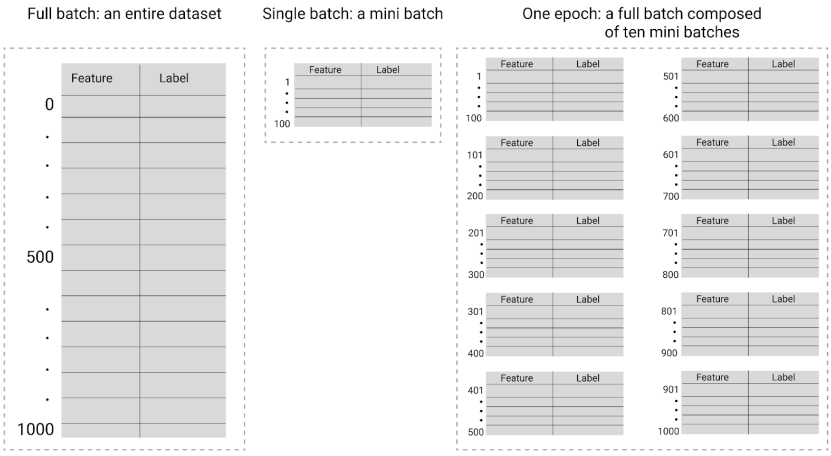


Figure 9: Full batch versus mini-batch

5 References

- Linear Regression - Google Developers
- Machine Learning Glossary - Google Developers