How to Normalize Data Between -1 and 1

Authored by

stats writer

November 21, 2025

Normalization, a critical procedure in data science and statistical modeling, involves transforming numerical features into a standardized range. One specific and highly effective method is scaling data to lie strictly between **-1** and **1**. This particular range scaling is often preferred in scenarios where maintaining the original distribution shape is crucial while ensuring numerical stability for algorithms like neural networks or optimization routines. When features possess vastly different scales--for instance, housing prices measured in millions and room counts measured in single digits--the algorithm might disproportionately weight the larger-scaled feature. Normalization addresses this inherent bias, ensuring all features contribute equally to the model's learning process, thereby stabilizing gradients and speeding up convergence.

Understanding Data Scaling and Normalization

Data preprocessing is the foundational step before any serious modeling begins. The choice of scaling technique depends heavily on the data's distribution and the specific requirements of the downstream algorithm. While **Z-score standardization** (which transforms data to have a mean of 0 and a standard deviation of 1) is common, it does not guarantee values fall within a specific boundary. In contrast, scaling to the **range**, typically achieved through a variation of Min-Max scaling, guarantees that the resulting scaled values are bounded. This bounded range is mathematically advantageous, especially when inputs must conform to constraints imposed by activation functions in deep learning, such as the hyperbolic tangent function (tanh), which also operates between -1 and 1.

The fundamental goal of this boundary-based normalization is to remove the influence of the arbitrary units of measurement. If one variable measures temperature in Kelvin (large positive numbers) and another measures humidity as a percentage (small numbers between 0 and 100), combining them directly in an optimization problem can lead to numerical instability. By mapping both datasets onto the identical range of -1 to 1, we ensure numerical compatibility. Furthermore, this technique is less susceptible to issues arising from features with highly non-linear relationships compared to standardization, providing a robust method for initial data preparation across various disciplines, including signal processing and image recognition.

It is important to note the difference between techniques. Standard Min-Max scaling usually maps values to the **range**. To achieve the **range**, a simple translation and additional scaling must be applied to the output of the standard transformation. This particular range centering around zero is often desirable because it helps the optimization process treat positive and negative deviations from the mean symmetrically, which can be critical for achieving faster convergence in iterative

algorithms like gradient descent. We must proceed cautiously, however, as like standard Min-Max scaling, this method is sensitive to extreme values or outliers, which can compress the bulk of the data into a very small portion of the normalized range.

The Mathematical Foundation: Min-Max Scaling for Range

To normalize the values in a dataset specifically to be between -1 and 1, we employ a modified version of the standard Min-Max scaling formula. This formula effectively calculates the relative position of a data point within its range, scales that relative position by two, and then shifts the result down by one, centering the data around zero. The transformation ensures that the minimum observed value in the original dataset is mapped precisely to -1, and the maximum observed value is mapped precisely to 1. All intermediate values are scaled linearly between these two boundaries.

The standard formula for scaling a value x_i to the range $$ is generally given by $x'_i = A + frac{(x_i - x_{min})(B - A)}{(x_{max} - x_{min})}$. In our specific case, we are targeting the range $A = -1$ and $B = 1$. Substituting these values into the general formula yields a simplified and highly practical relationship for achieving the desired bounds. This modified transformation is crucial for applications where input symmetry around zero is a strict requirement, such as training certain recurrent neural networks.

The precise formula used to map any arbitrary value x_i from the original dataset onto its corresponding normalized value z_i in the **range** is as follows:

zi = 2 * ((xi - xmin) / (xmax - xmin)) - 1

Let us carefully examine the components of this equation and their interpretation within the context of feature transformation. Understanding the role of each variable is essential for correct implementation and interpretation of the scaled data.

where:

zi: The ith normalized value in the dataset, which will always be between -1 and 1.
xi: The ith original value or observation point in the raw dataset.
xmin: The absolute minimum value found across the entire original dataset for that specific feature.
xmax: The absolute maximum value found across the entire original dataset for that specific feature.

Step-by-Step Calculation: Applying the Formula

To solidify the understanding of this normalization technique, we will work through a practical example using a small, representative dataset. Before applying the transformation, the absolute minimum (x_{min}) and maximum (x_{max}) values must be determined from the entire feature vector. These fixed bounds ensure that the scaling transformation is applied consistently across all data points, preventing data leakage if the process were performed on subsets. The integrity of the dataset bounds dictates the success of the scaling operation.

For example, suppose we have the following dataset of five observations. We must first identify the necessary parameters for the transformation based on these raw values.

Data Values
13
16
19
22
23
38
47
56
58
63
65
70
71

By inspecting the provided image containing the raw data values, we can clearly identify the extremal points. The minimum value in the dataset, **xmin**, is **13**, and the maximum value in the dataset, **xmax**, is **71**. These two parameters will remain constant throughout the normalization process for every single observation in this specific feature column. The range span, calculated as $(x_{max} - x_{min})$, is $71 - 13 = 58$. This span represents the denominator in the scaling component of our formula.

Visualizing the Transformation: Example Walkthrough

Let's demonstrate how specific data points are mapped to the new range using the derived minimum and maximum values. We start with the minimum value itself, 13, to confirm that it correctly maps to the lower bound of -1.

To normalize the first value of **13**, we apply the formula:

zi = 2 * ((xi - xmin) / (x_{max} - x_{min})) - 1 = 2 * ((13 - 13) / (71 - 13)) - 1 = **-1**

As expected, when the original value x_i equals the minimum value x_{min}, the fraction term $(x_i - x_{min}) / (x_{max} - x_{min})$ evaluates to zero, leading to the normalized value $z_i = 2 * (0) - 1 = -1$. Next, we examine a value close to the minimum, 16, to see how the scaling process proportionally translates a small shift in the original data.

To normalize the second value of **16**, we use the same formula:

zi = 2 * ((xi - x_{min}) / (x_{max} - x_{min})) - 1 = 2 * ((16 - 13) / (71 - 13)) - 1 = 2 * (3 / 58) - 1 ≈ 2 * (0.0517) - 1 ≈ **-0.897**

The slight increase from 13 to 16 results in a movement from -1 toward zero in the normalized space. This demonstrates the linear interpolation property of Min-Max scaling. We continue this process for the remaining values, ensuring meticulous calculation to maintain accuracy in the final scaled features.

To normalize the third value of **19**, we use the same formula:

zi = 2 * ((x_{i} - x_{min}) / (x_{max} - x_{min})) - 1 = 2 * ((19 - 13) / (71 - 13)) - 1 = 2 * (6 / 58) - 1 ≈ 2 * (0.1034) - 1 ≈ **-0.793**

By iterating this exact same formula across the entirety of the original dataset, every raw value is meticulously translated into its corresponding position within the new, bounded range of -1 to 1. This uniformity in application is what makes the resulting scaled dataset numerically stable and comparable across different features. The final output confirms that all transformed values successfully adhere to the required boundaries.

Data Values	Normalized
13	-1
16	-0.897
19	-0.793
22	-0.690
23	-0.655
38	-0.138
47	0.172
56	0.483
58	0.552
63	0.724
65	0.793
70	0.966
71	1

As illustrated in the resulting table above, each value in the normalized dataset is now confidently situated between **-1** and **1**. We can also verify that the maximum value of 71 in the original data would transform to exactly 1: $z_{i} = 2 times ((71 - 13) / (71 - 13)) - 1 = 2 times (1) - 1 = 1$. This confirms the integrity of the Min-Max transformation centered on zero.

Properties of Normalization

This specific normalization method guarantees several key properties that are beneficial for modeling, particularly in the realm of machine learning. Unlike standardization, which relies on the mean and standard deviation and thus is heavily influenced by the underlying data distribution, this Min-Max variant relies solely on the two most extreme points. Consequently, the transformation preserves the relative relationships and distances between data points, meaning that if point A was closer to point B than point C in the original scale, that relationship holds true in the normalized scale.

Using this normalization method, the following deterministic statements will always hold true, regardless of the complexity or size of the original dataset, provided the feature exhibits variation (i.e., $x_{max} neq x_{min}$):

The normalized value corresponding to the **minimum value** in the dataset will always be **-1**.

The normalized value corresponding to the **maximum value** in the dataset will always be **1**.
The normalized values for all other values residing strictly between the minimum and maximum will be strictly between **-1** and **1**.

Furthermore, scaling to the range inherently centers the data around zero. While the resulting mean may not be exactly zero (unlike Z-score standardization), the symmetric boundaries ensure that positive and negative values represent deviations above and below the midpoint of the original range, respectively. This centered structure is essential for algorithms that assume input symmetry, allowing for more efficient gradient updates during training.

When to Normalize Data: Context and Necessity

Normalization is typically necessitated when performing some type of analysis involving multiple variables that are measured on fundamentally different scales or possess highly disparate ranges. The purpose is to prevent features with large numerical magnitudes from dominating the objective function or distance calculations used by the algorithm. If we are comparing customer income (tens of thousands) with customer age (tens), an unscaled distance metric would be overwhelmingly determined by the income feature, effectively minimizing the contribution of age.

This equalization of range prevents one variable from being overly influential simply due to its unit of measurement. For instance, if one feature is measured in millimeters and a highly correlated feature is measured in kilometers, the numerical difference is immense. Algorithms relying on Euclidean distance, such as K-Nearest Neighbors (KNN) or K-Means Clustering, are particularly susceptible to this scale bias. Scaling to ensures that distance computations treat a unit change in one feature as equivalent to a unit change in another, regardless of their original metrics.

However, practitioners must consider that this specific normalization method is sensitive to outliers. If the dataset contains an extreme value far outside the typical distribution, the x_{max} (or x_{min}) parameter will be skewed by this outlier. This skew forces the entire body of normal data points to be compressed into a very narrow range near zero in the transformed space, potentially obscuring meaningful variations. If severe outliers are present, alternative methods like robust scaling or standardization might be more appropriate, or the outliers should be handled prior to scaling.

Alternatives to Normalization and Standardization

It is essential to recognize that the normalization method used here, mapping data to the range, is

just one viable option within the broader category of feature scaling techniques. Depending on the statistical properties of the data and the specific requirements of the modeling task, other scaling methods may offer superior performance or robustness. The choice is often an empirical one, requiring experimentation.

In some instances, it makes more sense to instead normalize variables to the **range** (standard Min-Max scaling), which is useful when all inputs must be non-negative, such as probability estimates or when the data is inherently positive, like pixel intensities in an image. Similarly, scaling data between **0 and 100** (percentile scaling) might be preferred for interpretability in business reports or specific psychological scaling applications. Furthermore, the aforementioned Z-score standardization, while not bounding the data, is highly effective for algorithms that assume normally distributed data and is far more robust to outliers than Min-Max approaches because it uses the mean and standard deviation, rather than the absolute minimum and maximum.

The selection of a scaling approach should therefore be guided by three main considerations: the mathematical requirements of the model (e.g., activation functions needing inputs in), the sensitivity of the technique to outliers, and the desired interpretability of the resulting scaled features. Experienced data scientists often test multiple scaling methods on the training data to determine which yields the best performance metrics for the final machine learning model.

Conclusion: Choosing the Right Scaling Method

Scaling data to the **range** provides a powerful and deterministic method for feature scaling, ensuring input symmetry around zero and strict boundaries, which are often non-negotiable prerequisites for specific neural network architectures. By applying the modified Min-Max formula, we guarantee that all values are comparable and equally weighted by downstream algorithms, effectively mitigating the scale bias inherent in raw datasets.

While this method is exceptionally clear and easy to implement, its susceptibility to extreme values mandates careful preprocessing and outlier detection before application. Mastery of data scaling, including knowing precisely when to use the range normalization versus Z-score standardization, is a hallmark of effective data science practice. By utilizing the provided formula and understanding its underlying assumptions, practitioners can ensure their data is optimally prepared for robust modeling.

The following tutorials explain how to perform other types of normalization: