Preprocessing Techniques Explained in Mathematical Terms

1. Cleaning The Signals:

Let the original signal having 15Hz Sample Rate with the following values,

Original Signal = [0.01, 0.03, 0.02, 0.06, 0.2, 0.5, 0.1, 0.04, 0.03, 0.5, 0.7, 0.05, 0.02, 0.03, 0.01]

Absolute Signal = [0.01, 0.03, 0.02, 0.06, 0.2, 0.5, 0.1, 0.04, 0.03, 0.5, 0.7, 0.05, 0.02, 0.03, 0.01]

Rolling Mean Calculation:

For the first element (0.01):

Window = [0.01, 0.03] (only two elements because it's at the boundary)

Rolling mean: (0.01 + 0.03)/2 = 0.02

For the second element (0.03):

Window = [0.01, 0.03, 0.02]

Rolling mean: (0.01+0.03+0.02)/3 = 0.02

For the third element (0.02):

Window = [0.03, 0.02, 0.06]

Rolling mean: (0.03+0.02+0.06)/3 = 0.037

For the fourth element (0.06):

Window = [0.02, 0.06, 0.2]

Rolling mean: (0.02+0.06+0.2)/3 = 0.093

For the fifth element (0.2):

Window = [0.06, 0.2, 0.5]

Rolling mean: (0.06+0.2+0.5)/3 = 0.2533

For the sixth element (0.5):

Window = [0.2, 0.5, 0.1]

Rolling mean: (0.2+0.5+0.1)/3 = 0.2667

For the seventh element (0.1):

Window = [0.5, 0.1, 0.04]

Rolling mean: (0.5+0.1+0.04)/3 = 0.2133

For the eighth element (0.04):

Window = [0.1, 0.04, 0.03]

Rolling mean: (0.1+0.04+0.03)/3 = 0.0567

For the ninth element (0.03):

Window = [0.04, 0.03, 0.5]

Rolling mean: (0.04+0.03+0.5)/3 = 0.19

For the tenth element (0.5):

Window = [0.03, 0.5, 0.7]

Rolling mean: (0.03+0.5+0.7)/3 = 0.41

For the eleventh element (0.7):

Window = [0.5, 0.7, 0.05]

Rolling mean: (0.5+0.7+0.05)/3 = 0.4167

For the twelfth element (0.05):

Window = [0.7, 0.05, 0.02]

Rolling mean: (0.7+0.05+0.02)/3 = 0.2567

For the thirteenth element (0.02):

Window = [0.05, 0.02, 0.03]

Rolling mean: (0.05+0.02+0.03)/3 = 0.0333

For the fourteenth element (0.03):

Window = [0.02, 0.03, 0.01]

Rolling mean: (0.02+0.03+0.01)/3 = 0.02

For the fifteenth element (0.01):

Window = [0.03, 0.01] (only two elements because it's at the boundary)

Rolling mean: (0.03+0.01)/2 = 0.02

Rolling Means = [0.02, 0.02, 0.037, 0.0933, 0.2533, 0.2667, 0.2133, 0.0567, 0.19, 0.41, 0.4167, 0.2567, 0.0333, 0.02, 0.02]

Masking/Filtering with Rolling Means

Let Mask Threshold = 0.1,

Mask = [False, False, False, False, True, True, True, False, True, True, True, True, False, False, False]

Abs Signal = [0.01, 0.03, 0.02, 0.06, 0.2, 0.5, 0.1, 0.04, 0.03, 0.5, 0.7, 0.05, 0.02, 0.03, 0.01]

Mask Apply:

0.01 → Mask: False → Discard

0.03 → Mask: False → Discard

0.02 → Mask: False → Discard

0.06 → Mask: False → Discard

0.2 → Mask: True → Keep

0.5 → Mask: True → Keep

0.1 → Mask: True → Keep

0.04 → Mask: False → Discard

0.03 → Mask: True → Keep

0.5 → Mask: True → Keep

0.7 → Mask: True → Keep

0.05 → Mask: True → Keep

0.02 → Mask: False → Discard

0.03 → Mask: False → Discard

0.01 → Mask: False → Discard

Cleaned Signal = [0.2, 0.5, 0.1, 0.03, 0.5, 0.7, 0.05]

2. Slowing Down Signals:

When slowing down by 0.6x, we increase the number of samples. To calculate this, we will add more points between the original samples by interpolating.

Clean Signal = [0.2, 0.5, 0.1, 0.03, 0.5, 0.7, 0.05]

Original Indices: [0, 1, 2, 3, 4, 5, 6]

New Indices (after slowing down): We multiply the original indices by 0.6 to get new points:

New Indices = [0, 0.6, 1.2, 1.8, 2.4, 3, 3.6, 4.2, 4.8, 5.4, 6]

From 0 to 6 we get 11 points compared to the original 7

For indices that are fractional (like 0.6, 1.2, etc.), we use linear interpolation between the two closest points.

Interpolation Formula:

Given two points,

y1 and y2, at indices x1 and x2, the interpolated value at point x is:

$y = y1 + (y2 - y1) \times (x - x1) / (x2 - x1)$

Step-by-Step Calculation (for Slowing Down):

Index 0: This corresponds exactly to index 0 of the original signals:

Value at index 0 = 0.2

Y = 0.2

Index 0.6: Interpolate between index 0 and index 1:

$Y = 0.2 + (0.5 - 0.2) \times (0.6 - 0) / (1 - 0) = 0.2 + 0.3 \times 0.6 = 0.38$

Index 1.2: Interpolate between index 1 and index 2:

$Y = 0.5 + (0.1 - 0.5) \times (1.2 - 1) / (2 - 1) = 0.5 + (-0.4) \times 0.2 = 0.42$

Index 1.8: Interpolate between index 1 and index 2:

$Y = 0.5 + (0.1 - 0.5) \times (1.8 - 1) / (2 - 1) = 0.5 + (-0.4) \times 0.8 = 0.18$

Index 2.4: Interpolate between index 2 and index 3:

$Y = 0.1 + (0.03 - 0.1) \times (2.4 - 2) / (3 - 2) = 0.1 + (-0.07) \times 0.4 = 0.072$

Index 3: This corresponds exactly to index 3 of the original signals:

Value at index 3 = 0.03

$Y = 0.03$

Index 3.6: Interpolate between index 3 and index 4:

$Y = 0.03 + (0.5 - 0.03) \times (3.6 - 3) / (4 - 3) = 0.03 + 0.47 \times 0.6 = 0.312$

Index 4.2: Interpolate between index 4 and index 5:

$Y = 0.5 + (0.7 - 0.5) \times (4.2 - 4) / (5 - 4) = 0.5 + 0.2 \times 0.2 = 0.54$

Index 4.8: Interpolate between index 4 and index 5:

$Y = 0.5 + (0.7 - 0.5) \times (4.8 - 4) / (5 - 4) = 0.5 + 0.2 \times 0.8 = 0.66$

Index 5.4: Interpolate between index 5 and index 6:

$Y = 0.7 + (0.05 - 0.7) \times (5.4 - 5) / (6 - 5) = 0.7 + (-0.65) \times 0.4 = 0.44$

Index 6: This corresponds exactly to index 6 of the original signals:

Value at index 6 = 0.05

$Y = 0.05$

Time-Stretched Signal (Slow),

Time Stretch Slow = [0.2, 0.38, 0.42, 0.18, 0.072, 0.03, 0.312, 0.54, 0.66, 0.44, 0.05]

Here, we have more points because the signal is being slowed down.

3. Speeding Up Signals:

We want to speed up the signal by 1.2x. This means that the signal will play 20% faster, and we will retain fewer samples (about 83.33% of the original).

Clean Signal = [0.2, 0.5, 0.1, 0.03, 0.5, 0.7, 0.05]

The original signal has 7 samples = [0, 1, 2, 3, 4, 5, 6]

After speeding up by a factor of 1.2, the new signal should only have around

$1.2 \times 7 \approx 5.83$ samples.

We round this to 6 samples for simplicity.

To speed up the signal, we need to determine the new positions for the samples. We divide the original indices by 1.2 to compress the timeline.

Original Indices: [0, 1, 2, 3, 4, 5, 6]

New Indices (dividing by 1.2),

New Indices = [0, 1, 2, 3, 4, 5, 6] / 1.2 = [0, 0.833, 1.667, 2.5, 3.333, 4.167, 5]

These new indices represent where we will sample from the original signal, but they don't align exactly with integer positions, so we need to interpolate to estimate the values at the fractional indices.

We need to retain approximately 83.33% of the original signal, reducing 7 samples to 6.

So, I've randomly skipped the 4.167 index to ensure that we only have 6 points, as expected.

New Indices = [0, 0.833, 1.667, 2.5, 3.333, 5]

Step-by-Step Calculation (For Speeding Up):

Index 0: This corresponds exactly to index 0 in the original signal, so the value remains:

Value at index 0 = 0.2

$Y = 0.2$

Index 0.833: Interpolate between index 0 and index 1:

$Y = 0.2 + (0.5 - 0.2) \times (0.833 - 0) / (1 - 0) = 0.2 + 0.3 \times 0.833 = 0.4499$

Index 1.667: Interpolate between index 1 and index 2:

$Y = 0.5 + (0.1 - 0.5) \times (1.667 - 1) / (2 - 1) = 0.5 + (-0.4) \times 0.667 = 0.2332$

Index 2.5: Interpolate between index 2 and index 3:

$Y = 0.1 + (0.03 - 0.1) \times (2.5 - 2) / (3 - 2) = 0.1 + (-0.07) \times 0.5 = 0.065$

Index 3.333: Interpolate between index 3 and index 4:

$Y = 0.03 + (0.5 - 0.03) \times (3.333 - 3) / (4 - 3) = 0.03 + 0.47 \times 0.333 = 0.1865$

Index 5: This corresponds exactly to index 5 in the original signal, so the value remains:

Value at index 5 = 0.7

$Y = 0.7$

Time-Stretched Signal (Fast),

Time Stretch Fast = [0.2, 0.4499, 0.2332, 0.065, 0.1865, 0.7]

Here, we have less points because the signal is being sped up.

4. Pitch Shift (Low)

Pitch shifting involves changing the perceived pitch of the audio signal by altering its frequency, without affecting the duration of the signal.

The general formula for pitch shifting is:

$f' = f \times 2^{n/12}$

Where,

f' is the new frequency (after pitch shift).

f is the original frequency.

n is the number of shift (positive for shifting up, negative for shifting down).

Pitch Shift Low by -5

Clean Signal = [0.2, 0.5, 0.1, 0.03, 0.5, 0.7, 0.05]

For a shift down by -5 semitones, the factor is:

Factor = $2^{-5/12} \approx 2^{-0.4167} \approx 0.749$

This means that each frequency component in the signal will be reduced to approximately 74.9% of its original value.

Pitch Shift Low = [0.2×0.749, 0.5×0.749, 0.1×0.749, 0.03×0.749, 0.5×0.749, 0.7×0.749, 0.05×0.749]

Pitch Shift Low = [0.1498, 0.3745, 0.0749, 0.0225, 0.3745, 0.5243, 0.0375]

This is the pitch-shifted low version of the signal, where the pitch is deeper, and the signal values are reduced to approximately 74.9% of their original magnitude.

This is the pitch-shifted high version of the signal, where the pitch is higher, and the signal values are increased to approximately 133.487% of their original magnitude.

5. Pitch Shift (High)

Pitch Shift High by +5

Clean Signal = [0.2, 0.5, 0.1, 0.03, 0.5, 0.7, 0.05]

For a shift up by +5 semitones, the factor is:

Factor = $2^{5/12} \approx 2^{0.4167} \approx 1.33487$

This means that each frequency component in the signal will be increased to approximately 133.487% of its original value.

Pitch Shift High = [0.2×1.33487, 0.5×1.33487, 0.1×1.33487, 0.03×1.33487, 0.5×1.33487, 0.7×1.33487, 0.05×1.33487]

Pitch Shift High = [0.267, 0.6674, 0.1335, 0.04005, 0.6674, 0.9344, 0.06674]

6. Noise Addition

Adding noise to a signal is a common technique in data augmentation, often used to simulate real-world conditions where perfect signals are rare. The idea is to add small random fluctuations (noise) to each value in the original signal.

We can add Gaussian noise (normally distributed noise) to the signal. The noise is usually defined by a mean μ and a standard deviation σ.

In this case, we'll generate Gaussian noise with:

$\mu = 0$ (mean of 0, meaning no systematic bias in one direction).

$\sigma = 0.001$ (a small standard deviation, meaning the noise will only slightly affect the signal).

Clean Signal = [0.2, 0.5, 0.1, 0.03, 0.5, 0.7, 0.05]

We will generate random noise for each element in the signal using the formula for Gaussian noise:

Noise = $\sigma \times N(0,1)$

Where, $N(0,1)$ is a normally distributed random value (mean 0, standard deviation 1), and $\sigma = 0.001$.

Let’s assume we generate the following random values for $N(0,1)$ for each signal element (rounded for clarity):

$N1 = 0.5$

$N2 = -1.2$

$N3 = 0.8$

$N4 = 1.1$

$N5 = -0.7$

$N6 = 0.4$

$N7 = -1.0$

Now we multiply each random value by $\sigma = 0.001$ to get the noise values:

Noise = $[0.001\times0.5, 0.001\times-1.2, 0.001\times0.8, 0.001\times1.1, 0.001\times-0.7, 0.001\times0.4, 0.001\times-1.0]$

Resulting noise:

Noise = $[0.0005, -0.0012, 0.0008, 0.0011, -0.0007, 0.0004, -0.001]$

Now, we add the generated noise to each corresponding element of the original signal:

noisy signal = clean signal + noise

Performing the calculation:

$0.2 + 0.0005 = 0.2005$

$0.5 - 0.0012 = 0.4988$

$0.1 + 0.0008 = 0.1008$

$0.03 + 0.0011 = 0.0311$

$0.5 - 0.0007 = 0.4993$

$0.7 + 0.0004 = 0.7004$

$0.05 - 0.001 = 0.049$

The resulting noisy signal is:

noisy signal = [0.2005, 0.4988, 0.1008, 0.0311, 0.4993, 0.7004, 0.049]