Gaussian Distributions: Sum and Product

Introduction

In probability theory, the sum and product of Gaussian distributions exhibit interesting properties. In this notebook, we explore the mathematical concepts and visualize the distributions using Python.

Sum of Gaussian Distributions

Consider two independent Gaussian distributions (X \sim \mathcal{N}(\mu_1, \sigma_1^2)) and (Y \sim \mathcal{N}(\mu_2, \sigma_2^2)). The sum of these distributions, (Z = X + Y), follows another Gaussian distribution with the following characteristics:

[\mu_z = \mu_1 + \mu_2]

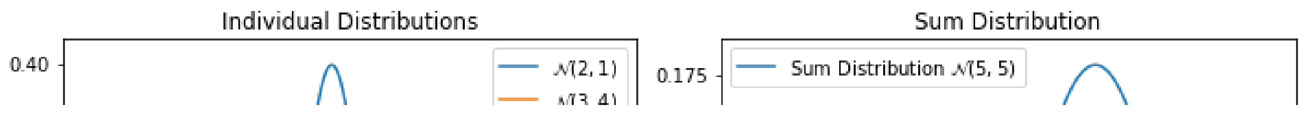
[\sigma_z^2 = \sigma_1^2 + \sigma_2^2]

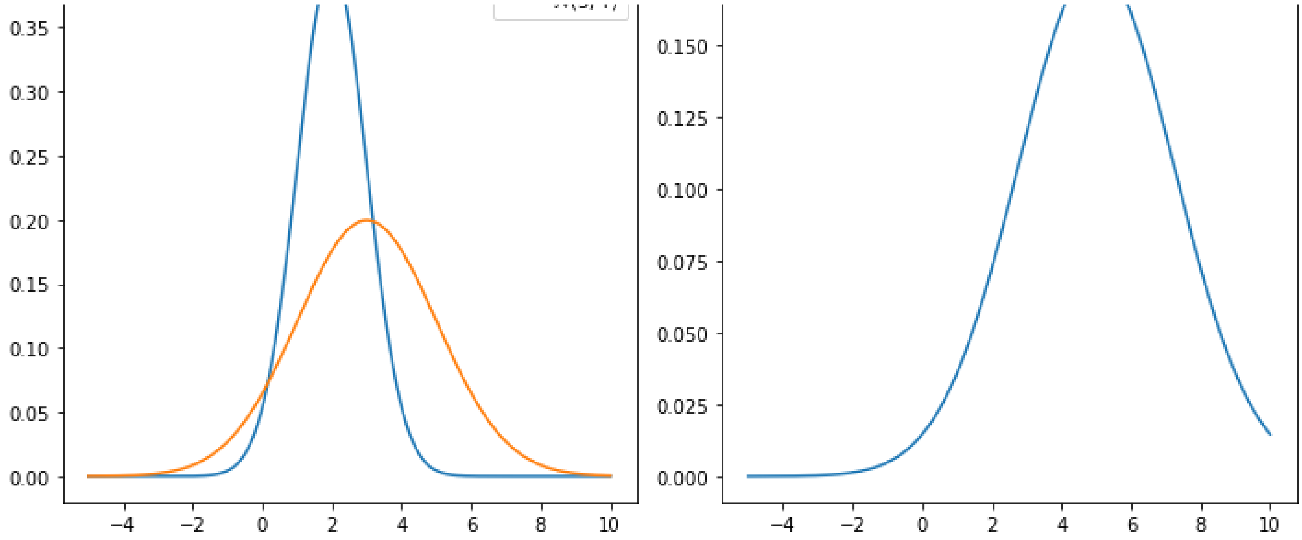
Let's visualize this concept with Python and matplotlib.

```python

Code for visualizing the sum of Gaussian distributions

```

In [1]:

~~~python
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm

# Parameters for the distributions
mu1, sigma1 = 2, 1
mu2, sigma2 = 3, 2  # using sqrt(4) for standard deviation

# Generate data points for the distributions
x = np.linspace(-5, 10, 1000)
pdf_x = norm.pdf(x, mu1, sigma1)
pdf_y = norm.pdf(x, mu2, sigma2)

# Plot the individual distributions
plt.figure(figsize=(10, 5))

plt.subplot(1, 2, 1)
plt.plot(x, pdf_x, label=r'$\mathcal{N}(2, 1)$')
plt.plot(x, pdf_y, label=r'$\mathcal{N}(3, 4)$')
plt.title('Individual Distributions')
plt.legend()

# Plot the sum distribution
plt.subplot(1, 2, 2)
pdf_sum = norm.pdf(x, mu1 + mu2, np.sqrt(sigma1**2 + sigma2**2))
plt.plot(x, pdf_sum, label=r'Sum Distribution $\mathcal{N}(5, 5)$')
plt.title('Sum Distribution')
plt.legend()

plt.tight_layout()
plt.show()
~~~





Product of Gaussian Distributions (Continued)

Now, let's consider the product of two independent Gaussian distributions (X \sim \mathcal{N}(\mu_1, \sigma_1^2)) and (Y \sim \mathcal{N}(\mu_2, \sigma_2^2)). The product distribution involves convolution, and the resulting distribution has the following properties:

[\mu_z = \mu_1 + \mu_2]

[\sigma_z^2 = \sigma_1^2 \cdot \sigma_2^2]

The convolution operation is more complex, and visualizing it directly can be challenging. We will use numerical methods to demonstrate this concept.

```python

Code for visualizing the product of Gaussian distributions (convolution)

...

In [2]:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm

# Parameters for the distributions
mu1, sigma1 = 2, 1
mu2, sigma2 = 3, 2

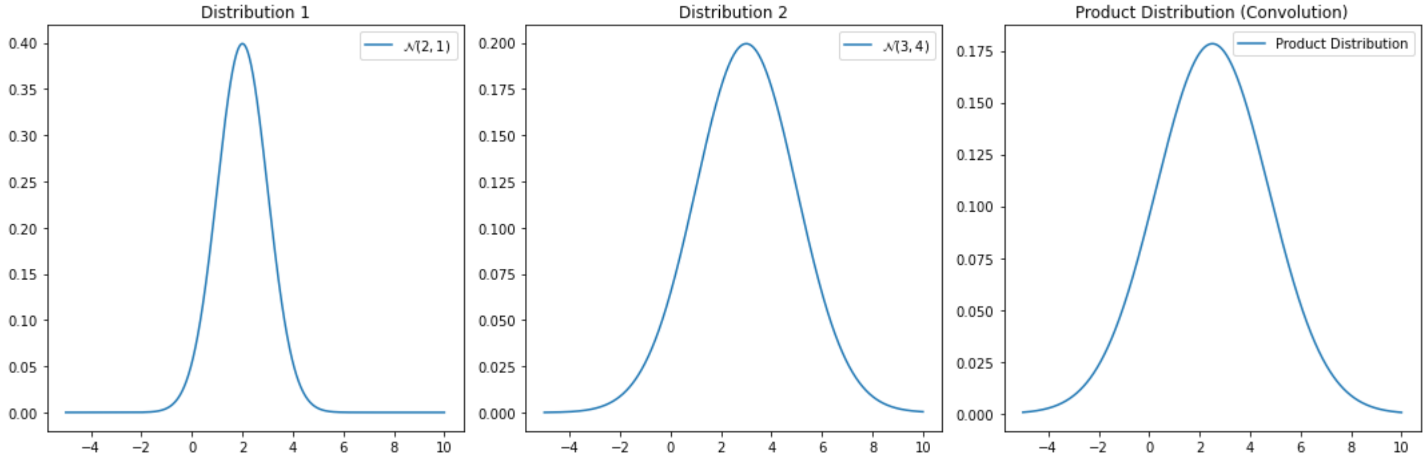
# Generate data points for the distributions
x = np.linspace(-5, 10, 1000)
pdf_x = norm.pdf(x, mu1, sigma1)
pdf_y = norm.pdf(x, mu2, sigma2)

# Plot the individual distributions
plt.figure(figsize=(15, 5))

plt.subplot(1, 3, 1)
plt.plot(x, pdf_x, label=r'$\mathcal{N}(2, 1)$')
plt.title('Distribution 1')
plt.legend()

plt.subplot(1, 3, 2)
plt.plot(x, pdf_y, label=r'$\mathcal{N}(3, 4)$')
plt.title('Distribution 2')
plt.legend()
```

```python
# Plot the product distribution (convolution)
pdf_product = np.convolve(pdf_x, pdf_y, mode='same') / np.sum(pdf_x)
plt.subplot(1, 3, 3)
plt.plot(x, pdf_product, label='Product Distribution')
plt.title('Product Distribution (Convolution)')
plt.legend()

plt.tight_layout()
plt.show()
```



In [ ]: