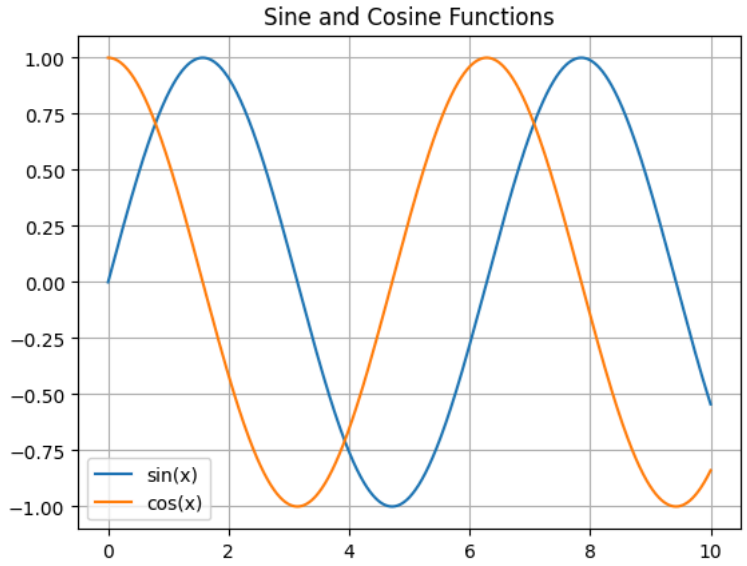
```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(0, 10, 1000)
plt.plot(x, np.sin(x), label='sin(x)')
plt.plot(x, np.cos(x), label='cos(x)')
plt.title("Sine and Cosine Functions")
plt.legend()
plt.grid(True)
plt.show()
```




```
import numpy as np
import matplotlib.pyplot as plt

# Sentence and settings
sentence = ["I", "like", "pizza"]
positions = list(range(len(sentence)))
d_model = 4  # embedding dimension
```

```
# Positional encoding function
def positional_encoding(pos, d_model):
    pe = np.zeros(d_model)
    for i in range(d_model):
        div_term = 10000 ** (2 * (i // 2) / d_model)
        if i % 2 == 0:
            pe[i] = np.sin(pos / div_term)
        else:
            pe[i] = np.cos(pos / div_term)
    return pe
```

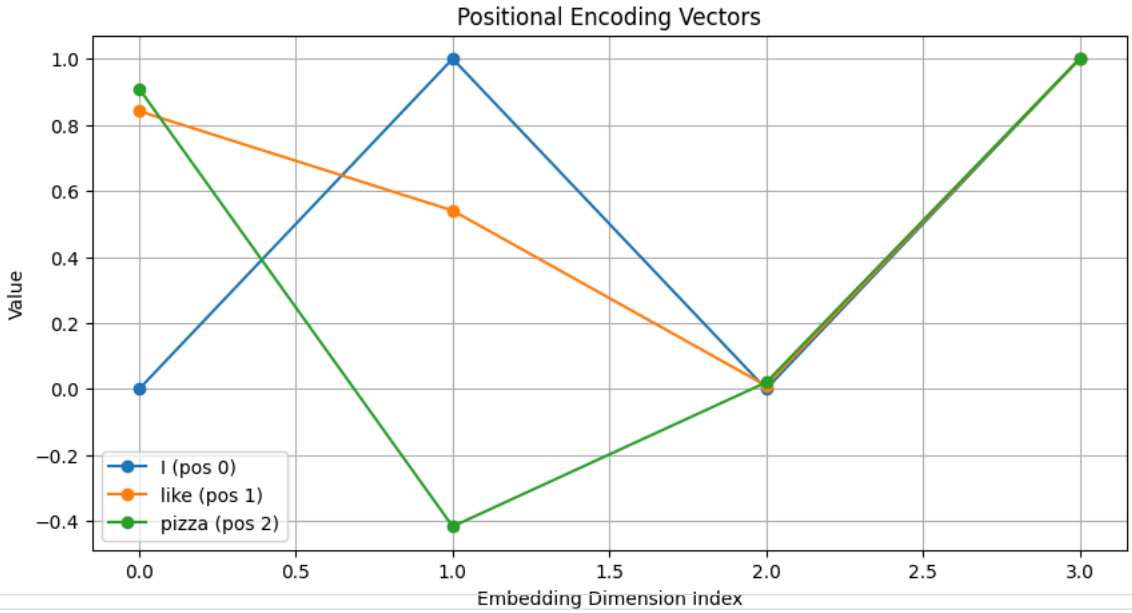
```
# Generate positional encodings
positional_encodings = np.array([positional_encoding(pos, d_model) for pos in positions])

# Print positional encoding vectors
for word, encoding in zip(sentence, positional_encodings):
    print(f"{word}: {encoding}")
```

```
I: [0. 1. 0. 1.]
like: [0.84147098 0.54030231 0.00999983 0.99995   ]
pizza: [ 0.90929743 -0.41614684  0.01999867  0.99980001]
```

```
# Plotting the positional encoding vectors
plt.figure(figsize=(10, 5))
for i, encoding in enumerate(positional_encodings):
    plt.plot(range(d_model), encoding, marker='o', label=f"{sentence[i]} (pos {i})")

plt.title("Positional Encoding Vectors")
plt.xlabel("Embedding Dimension Index")
plt.ylabel("Value")
plt.legend()
plt.grid(True)
plt.show()
```



Start coding or generate with AI.