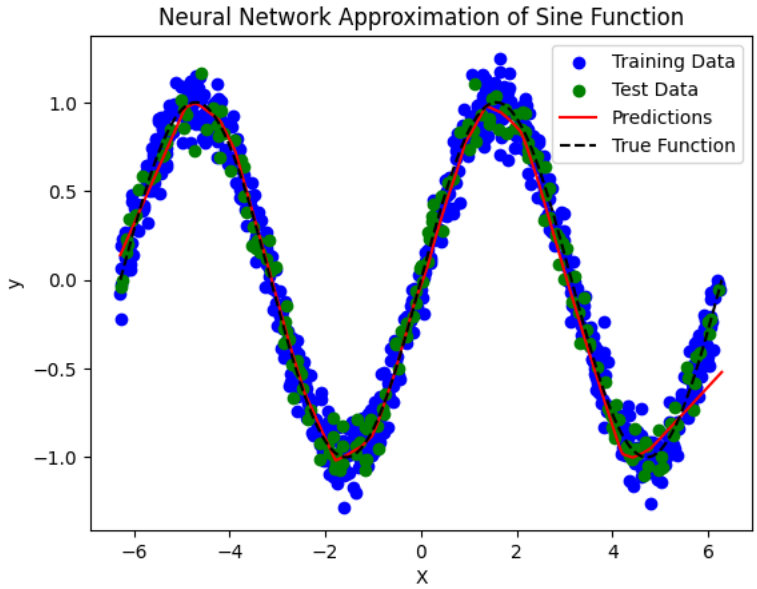
```
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
from sklearn.model_selection import train_test_split

# Generate synthetic data for a sine function
X = np.random.uniform(-2*np.pi, 2*np.pi, 1000)
y = np.sin(X) + np.random.normal(0, 0.1, 1000)

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Define and train a neural network model
model = tf.keras.Sequential([
    tf.keras.layers.Dense(64, activation='relu', input_shape=(1,)),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(1)
])
model.compile(optimizer='adam', loss='mean_squared_error')
model.fit(X_train, y_train, epochs=100, batch_size=32, validation_data=(X_test, y_test), verbose=0)

# Evaluate the model on the test set
loss = model.evaluate(X_test, y_test)
print(f'Mean Squared Error on Test Set: {loss:.4f}')

# Generate predictions using the trained model
X_pred = np.linspace(-2*np.pi, 2*np.pi, 1000)
y_pred = model.predict(X_pred)

# Plot the true function, training data, and predictions
plt.scatter(X_train, y_train, color='blue', label='Training Data')
plt.scatter(X_test, y_test, color='green', label='Test Data')
plt.plot(X_pred, y_pred, color='red', label='Predictions')
plt.plot(X_pred, np.sin(X_pred), 'k--', label='True Function')
plt.title('Neural Network Approximation of Sine Function')
plt.xlabel('X')
plt.ylabel('y')
plt.legend()
plt.show()
```

```
7/7 [==============================] - 0s 3ms/step - loss: 0.0139
Mean Squared Error on Test Set: 0.0139
32/32 [==============================] - 0s 2ms/step
```



create an application centered around the genrealization approximation of functions within neural network and develop a computational element to explore and implement techniques that enhance the generalization capabilities of neural networks perticularly in approximation complex functions

create an application centered around the genrealization approximati
functions within neural network and develop a computational element
and implement techniques that enhance the generalization capabilitie
networks perticularly in approximation complex functions