In [ ]:
```
import zipfile
import os
zip_path = "mnist.zip"
extract_path = "mnist_data"
with zipfile.ZipFile(zip_path, 'r') as zip_ref:
    zip_ref.extractall(extract_path)
print("Extraction complete.")
```

In [ ]:
```
import os
extracted_files_dir = 'extracted_files'
files = os.listdir(extracted_files_dir)
print("Extracted files:", files)
```

In [ ]:
```
import os

extracted_files = os.listdir('data')
print(extracted_files)
```

In [ ]:
```
import os
print("Current Working Directory:", os.getcwd())
```

In [ ]:
```
import zipfile

zip_path = "t10k-images.idx3-ubyte.zip"
extract_dir = "./extracted_files"

with zipfile.ZipFile(zip_path, "r") as zip_ref:
    zip_ref.extractall(extract_dir)

print("Extracted files:", os.listdir(extract_dir))
```

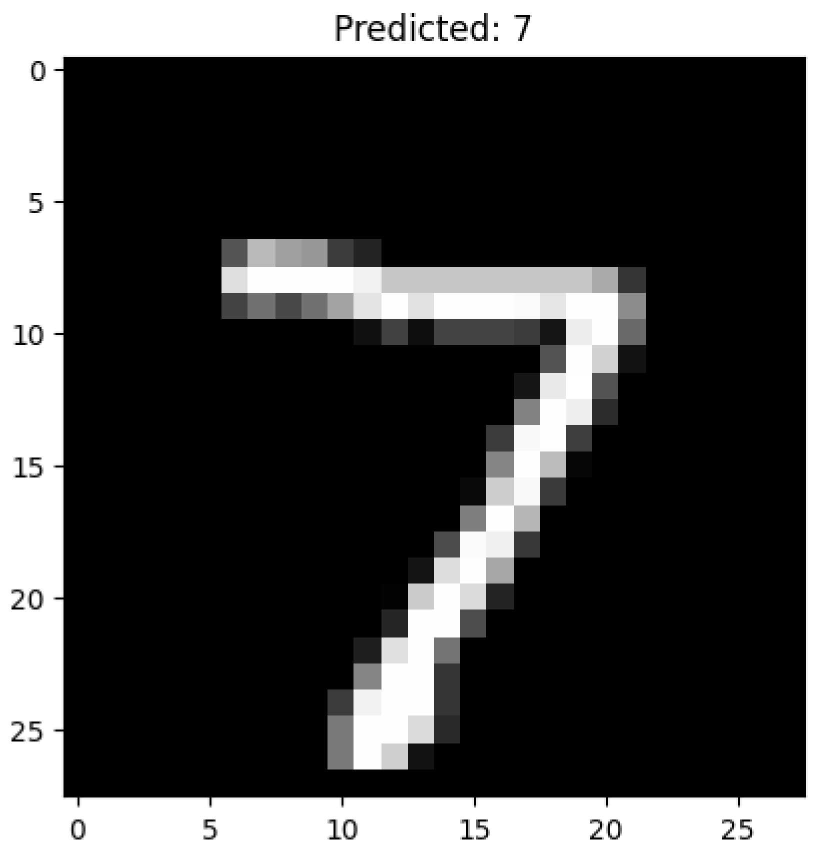
In [ ]:
```
filename = "./extracted_files/t10k-images.idx3-ubyte"
images = load_idx_images(filename)
```

In [22]:
```
import numpy as np
import struct
import matplotlib.pyplot as plt
def load_images(path):
    with open(path, "rb") as f:
        _, num, rows, cols = struct.unpack(">IIII", f.read(16))
        return np.frombuffer(f.read(), dtype=np.uint8).reshape(num, rows * cols)
def load_labels(path):
    with open(path, "rb") as f:
        _, num = struct.unpack(">II", f.read(8))
        return np.frombuffer(f.read(), dtype=np.uint8)
def sigmoid(x): return 1 / (1 + np.exp(-x))
def sigmoid_deriv(x): return x * (1 - x)
X = load_images("./extracted_files/t10k-images.idx3-ubyte") / 255.0
y = np.eye(10)[load_labels("./extracted_files/t10k-labels.idx1-ubyte")]
input_neurons, hidden_neurons, output_neurons = 784, 128, 10
np.random.seed(42)
W1, W2 = np.random.randn(input_neurons, hidden_neurons) * 0.01, np.random.randn(
def forward(X_batch):
    h_in = np.dot(X_batch, W1)
    h_out = sigmoid(h_in)
    o_in = np.dot(h_out, W2)
    o_out = sigmoid(o_in)
    return o_out, h_out
def train(X, y, epochs=100, lr=0.1, batch_size=128):
    global W1, W2
    num_samples = X.shape[0]
    for epoch in range(epochs):
        loss = 0
        indices = np.random.permutation(num_samples)
        X_shuffled = X[indices]
        y_shuffled = y[indices]
        for i in range(0, num_samples, batch_size):
            X_batch = X_shuffled[i:i+batch_size]
            y_batch = y_shuffled[i:i+batch_size]
            out, h_out = forward(X_batch)
            err = y_batch - out
            loss += np.sum(err ** 2)
            d_out = err * sigmoid_deriv(out)
            d_hid = np.dot(d_out, W2.T) * sigmoid_deriv(h_out)
            W2 += np.dot(h_out.T, d_out) * lr
            W1 += np.dot(X_batch.T, d_hid) * lr

        if epoch % 100 == 0:
            print(f"Epoch {epoch}, Loss: {loss:.4f}")
train(X, y, epochs=100, lr=0.1, batch_size=32)
test_img = X[0]
pred, _ = forward(test_img.reshape(1, -1))

plt.imshow(test_img.reshape(28, 28), cmap="gray")
plt.title(f"Predicted: {np.argmax(pred)}")
plt.show()
```

```
Epoch 0, Loss: 9616.6335
```


Predicted: 7