In [1]:

```
import torch
import torch.nn as nn

class TemporalModel(nn.Module):
    def __init__(self, input_size, hidden_size, num_layers, output_size):
        super(TemporalModel, self).__init__()
        self.hidden_size = hidden_size
        self.num_layers = num_layers
        self.lstm = nn.LSTM(input_size, hidden_size, num_layers, batch_firs
        self.fc = nn.Linear(hidden_size, output_size)

    def forward(self, x):
        # Initialize hidden state with zeros
        h0 = torch.zeros(self.num_layers, x.size(0), self.hidden_size).to(x
        # Initialize cell state
        c0 = torch.zeros(self.num_layers, x.size(0), self.hidden_size).to(x
        # We need to detach as we are doing truncated backpropagation throu
        # If we don't, we'll backprop all the way to the start even after g
        out, (hn, cn) = self.lstm(x, (h0.detach(), c0.detach()))
        # Decode the hidden state of the last time step
        out = self.fc(out[:, -1, :])
        return out

# Define the model
input_size = 10  # Number of features
hidden_size = 50  # Number of features in hidden state
num_layers = 2  # Number of stacked LSTM layers
output_size = 1  # Number of output classes

model = TemporalModel(input_size, hidden_size, num_layers, output_size)

# Example of input data (batch_size, sequence_length, input_size)
x = torch.randn(32, 5, input_size)

# Forward pass
output = model(x)
print(output)
```

```
tensor([[-0.1885],
        [-0.1793],
        [-0.1829],
        [-0.1787],
        [-0.1785],
        [-0.1834],
        [-0.1791],
        [-0.1843],
        [-0.1762],
        [-0.1864],
        [-0.1814],
        [-0.1858],
        [-0.1789],
        [-0.1698],
        [-0.1667],
        [-0.1912],
        [-0.1683],
        [-0.1766],
        [-0.1665],
        [-0.1832],
        [-0.1766],
        [-0.1826],
        [-0.1848],
        [-0.1892],
        [-0.1802],
        [-0.1824],
        [-0.1773],
        [-0.1891],
        [-0.1756],
        [-0.1823],
        [-0.1777],
        [-0.1706]], grad_fn=<AddmmBackward0>)
```

In [4]:

```
!pytest test_temporal_model.py -s
```

```
============================= test session starts =======================
======
platform win32 -- Python 3.8.18, pytest-7.4.0, pluggy-1.0.0
rootdir: C:\Users\kesha\workspace\pythonProject
plugins: anyio-3.5.0
tensor([[-0.0330],
        [-0.0579],
        [-0.0499],
        [-0.0469],
        [-0.0670],
        [-0.0418],
        [-0.0511],
        [-0.0337],
        [-0.0547],
        [-0.0659],
        [-0.0505],
        [-0.0654],
        [-0.0472],
        [-0.0695],
        [-0.0350],
        [-0.0400],
        [-0.0438],
        [-0.0431],
        [-0.0624],
        [-0.0533],
        [-0.0381],
        [-0.0473],
        [-0.0569],
        [-0.0597],
        [-0.0624],
        [-0.0555],
        [-0.0481],
        [-0.0578],
        [-0.0512],
        [-0.0315],
        [-0.0453],
        [-0.0527]], grad_fn=<AddmmBackward0>)
collected 1 item

test_temporal_model.py .

============================== 1 passed in 1.67s ========================
======
```

```
pytest
```

In [ ]: