```python
import torch
from torch.nn import Module, Linear
from torch.nn.functional import relu


class Model(Module):
    """An implementation of torch.nn.Module.

    Args:
        Module (Class): generic pytroch model class.
    """

    def __init__(self, in_shape: torch.Size, num_classes: int):
        """Initialize the model

        Args:
            in_shape (torch.Size): the shape of input.
            num_classes (int, optional): number of output classes.
        """
        super(Model, self).__init__()

        # Parameters
        self.in_features = torch.prod(torch.tensor(in_shape[1:]))
        self.num_classes = num_classes

        # Define layers
        self.fc0 = Linear(self.in_features, 32)
        self.fc1 = Linear(32, 32)
        self.fc2 = Linear(32, self.num_classes)

    def forward(self, x: torch.Tensor) -> torch.Tensor:
        """Feed data through the model.

        Args:
            x (torch.Tensor): data.

        Returns:
            torch.Tensor: label.
        """
        x = relu(self.fc0(x))
        x = relu(self.fc1(x))
        x = self.fc2(x)
        return x
```