2025 USA-NA-AIO Round 1, Problem 2, Part 11

USAAIO Mar 2025

Part 11 (10 points, coding task)

It is known by math that the combination of several linear layers can still be seen as an linear layer, so we can add some non-linear activation functions, such as ReLU, in between to get better effect.

Multi-Layer Perceptron (MLP), is such a neural network composed of multiple fully connected layers with non-linear activations, commonly used in deep learning.

Please define a class called `My_MLP_Model` that subclasses `nn.Module` and works in the following ways:

- The architecture consists of two hidden layers and one output layer.
- Each hidden layer consists of an affine transformation module and a ReLU activiation module.
- Each affine transformation module shall be initialized with the build-in class `nn.Linear`.
- Each ReLU activiatio nmodule shall be initialized with your self-defined class `My_ReLU`.

USAAIO Mar 2025

```
### WRITE YOUR SOLUTION HERE ###

class My_MLP_Model(nn.Module):
    def __init__(self, in_features, hidden_features1, hidden_features2, out_featu
        super().__init__()
        self.in_features = in_features
```

Skip to main content

```
            self.hidden_features1 = hidden_features1
            self.hidden_features2 = hidden_features2
            self.out_features = out_features
            self.seq = nn.Sequential(
                nn.Linear(in_features, hidden_features1),
                My_ReLU(),
                nn.Linear(hidden_features1, hidden_features2),
                My_ReLU(),
                nn.Linear(hidden_features2, out_features)
            )

        def forward(self, x):
            x = self.seq(x)
            return x

    """ END OF THIS PART """
```

Doughwhee **Oct 2025**

Does this work?

```
class My_MLP_Model(nn.Module):
  def __init__(self, in_features, hidden1_features, hidden2_features, out_feature
    super().__init__()
    self.fc1 = nn.Linear(in_features, hidden1_features)
    self.relu1 = My_ReLU()
    self.fc2 = nn.Linear(hidden1_features, hidden2_features)
    self.relu2 = My_Relu()
    self.fc3 = nn.Linear(hidden2_features, out_features)

  def forward(self, X):
    X = self.relu1(self.fc1(X))
    X = self.relu2(self.fc2(X))
```

Skip to main content

```
        X = self.fc3(X)
        return X
```

Related topics

Topic	Replies	Activity
2025 USA-NA-AIO Round 1, Problem 2, Part 5	1	Mar 2025
2026 USAAIO Round 1 Sample problems, Problem 10	1	Jan 14
2025 USA-NA-AIO Round 1, Problem 2, Part 10	1	Mar 2025
2025 USA-NA-AIO Round 1, Problem 2, Part 3	1	Mar 2025
2025 USA-NA-AIO Round 1, Problem 2, Part 9	4	Oct 2025