

2025 USA-NA-AIO Round 1, Problem 2, Part 9

 Topics

 More

 Categories

General

Site Feedback

All categories

USAAIO 

Mar 2025

Part 9 (5 points, coding task)

This question follows Part 6.

In this part, you are asked to program with PyTorch, not NumPy.

Build a deep neural network class named as Symmetric_Linear_Model that meets the modifications imposed in Part 6.

USAAIO 

Mar 2025

WRITE YOUR SOLUTION HERE

```
class Symmetric_Linear_Model(nn.Module):
    def __init__(self, in_features, hidden_features):
        super().__init__()
        self.in_features = in_features
        self.hidden_features = hidden_features
        self.linear0 = nn.Linear(in_features, hidden_features, bias = False)

    def forward(self, x):
        x = self.linear0(x)
        x = torch.sum(self.linear0.weight * x.reshape(*x.shape, 1), dim = -2)
        return x
```

"" END OF THIS PART """



[Skip to main content](#)

counter**Oct 2025**

This is my solution, does it also work?

```
class Symmetric_Linear_Model(nn.Module):
    def __init__(self, in_features, hidden_features):
        super().__init__()
        self.weights = nn.Parameter(torch.randn(hidden_features, in_features))

    def forward(self, x):
        return self.weights.T @ self.weights @ x
```

Doughwhee**Oct 2025**

It looks fine, maybe x should premultiply $\text{self.weights.T}@\text{self.weights}$? I think that allows it to handle batches correctly.

◆ Related topics

[Skip to main content](#)

Topic	Replies	Activity
-------	---------	----------

Topic	Replies	Activity
2025 USA-NA-AIO Round 1, Problem 2, Part 5	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 6	2	Dec 2025
2025 USA-NA-AIO Round 1, Problem 2, Part 11	2	Oct 2025
2026 USAAIO Round 1 Sample problems, Problem 11	1	Jan 14

 Powered by Discourse