2025 USA-NA-AIO Round 1, Problem 2, Part 3

USAAIO Mar 2025

Part 3 (10 points, coding task)

In this part, you are asked to build an affine transformation module from scratch by using **NumPy, NOT PyTorch or TensorFlow**.

Define such a class as `My_Linear_NumPy`.

- Attributes
  - `in_features`: Number of input features
  - `out_features`: Number of output features
  - `weight`: This refers to matrix $\mathbf{W}$ in Part 1. The shape is `(out_features, in_features)`.
  - `bias`: This refers to vector $\mathbf{b}$ in Part 1. The shape is `(out_features,)`.
  - `random_seed`: The NumPy random seed number used to generate initial values of `weight` and `bias`.
- Method `__init__`:
  - To initialize an object in this class, you need to specify `in_features` and `out_features`.
  - You may initialize the object by specifying a value for `random_seed`. If it is not specified, then its default value is 42.

Skip to main content

  - The initial values of `weight` and `bias` are random that follow standard normal distributions generated with the seed number attribute `random_seed`.

- Method `forward`:

  - Input `x`: numpy array with shape `(n_0, n_1, ..., n_{d-1}, in_features)` with an arbitrary dimension $d = 0, 1, \cdots$.

  - Output `y`: numpy array with shape `(n_0, n_1, ..., n_{d-1}, out_features)`.

  - The affine transformation works in a way that given the first d indices in `x` and `y`, it does affine transformation along the last axis of `x` and `y`.

- **Do not use any loop in your code.**

USAAIO **Mar 2025**

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

class My_Linear_NumPy:
    def __init__(self, in_features, out_features, random_seed=42):
        self.in_features = in_features
        self.out_features = out_features
        self.random_seed = random_seed
        np.random.seed(self.random_seed)
        self.weight = np.random.randn(self.out_features, self.in_features)
        self.bias = np.random.randn(self.out_features)

    def forward(self, x):
        return np.sum(x[..., np.newaxis] * self.weight.T, axis = -2) + self.bias

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

Skip to main content

Related topics

Topic	Replies	Activity
2025 USA-NA-AIO Round 1, Problem 2, Part 1	2	Dec 2025
2025 USA-NA-AIO Round 1, Problem 2, Part 4	1	Mar 2025
2025 USA-NA-AIO Round 1, Problem 2, Part 5	1	Mar 2025
2025 USA-NA-AIO Round 1, Problem 2, Part 9	4	Oct 2025
2025 USA-NA-AIO Round 1, Problem 2, Part 7	1	Mar 2025