Cyberdyne System's Neural Networks

Scenario

Having acquired several robotics companies in the past few weeks, Google has created a division of autonomous robotics research, named Cyberdyne Systems. The Google building in Boulder has been completely remodeled to work on these research projects. You are part of a team who has been tasked with developing training software for room temperature superconducting artificial neural network systems. Primarily, you are to find an efficient way for a new neural network to be created based on the experience gained by an existing neural network.

Problem

Your task is to create a mock neural network class. This class consists of a private attribute *weights*, which is simply a list of doubles, and two public methods, *learn()* and *respond().* Learn will modify the *weights* (for your implementation, just add small random values to each individual weight), and *respond* generates some results (for your implementation, just return the sum of the weights).

The last task is to easily enable a new instance of your neural network class to be created based on the experience of an existing neural network. Basically, this means that the newly created neural network has the same values for the *weights* as the existing neural network. You may extend the API for neural network as you see fit, however, you cannot change the accessibility of *weights* (it must remain private). To demonstrate, create a short simulation which creates an instance of a neural network, and call the *learn* method a few times. Then, create a new neural network based on the first neural network, so that both provide the same value when *respond* is called (until *learn* is called again on either instance).

Deliverables

1. Identify the design pattern you used to solve this problem, and the participants (i.e., the roles each class takes).
2. An implementation in a language of your choice.
3. A class diagram of your solution (including existing classes), so future developers can easily see how to work with your solution.