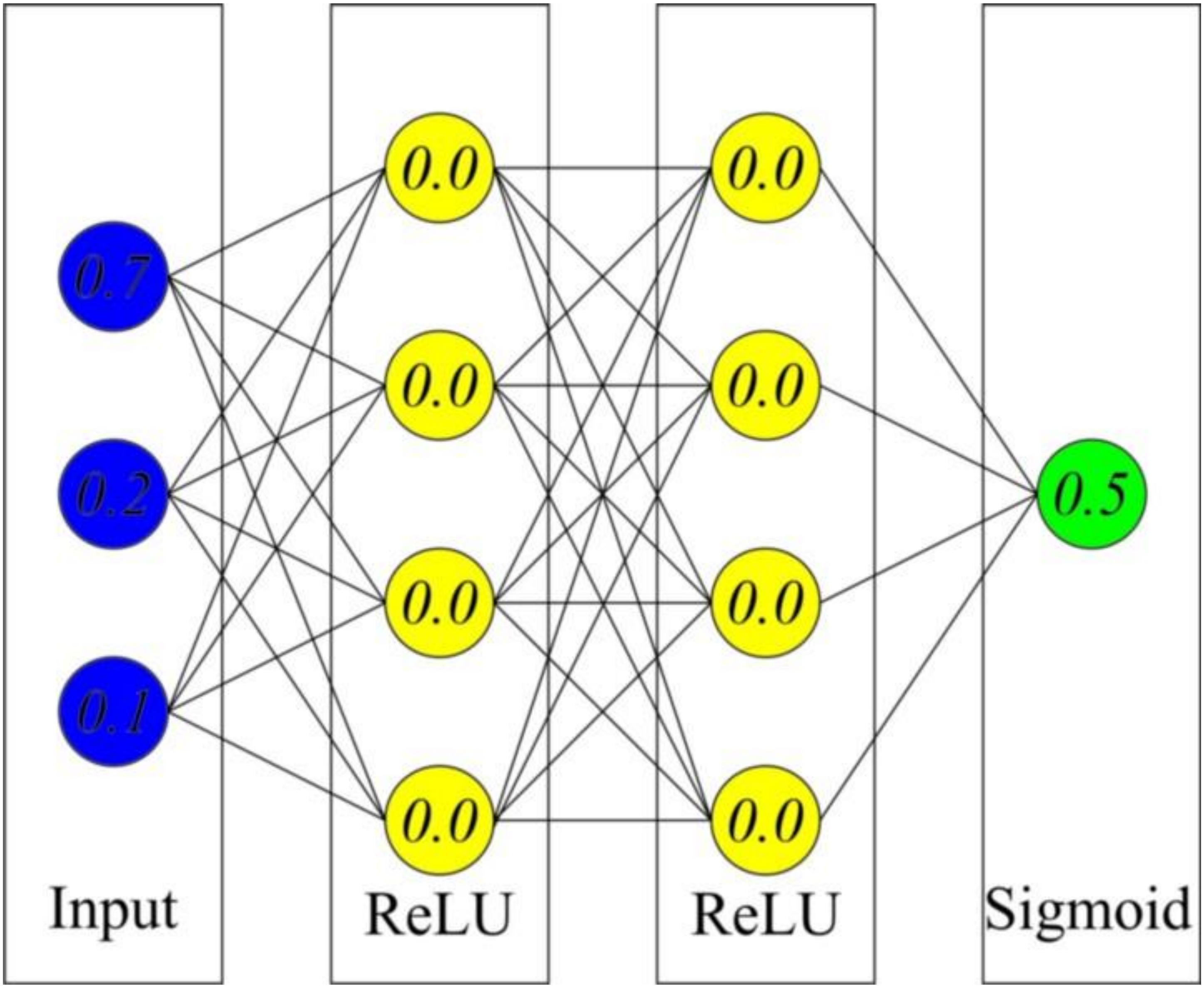
Exercise 2 :

In the neural network, the models that we design must act in such a way that they first receive data as input, and finally display the output by analyzing and processing them in hidden layers, and finally, with backward propagation, they can calculate the weights. update itself to become an ideal model for data analysis.

As a result, the initialization of the weights will have a very important effect on the train process and model improvement.

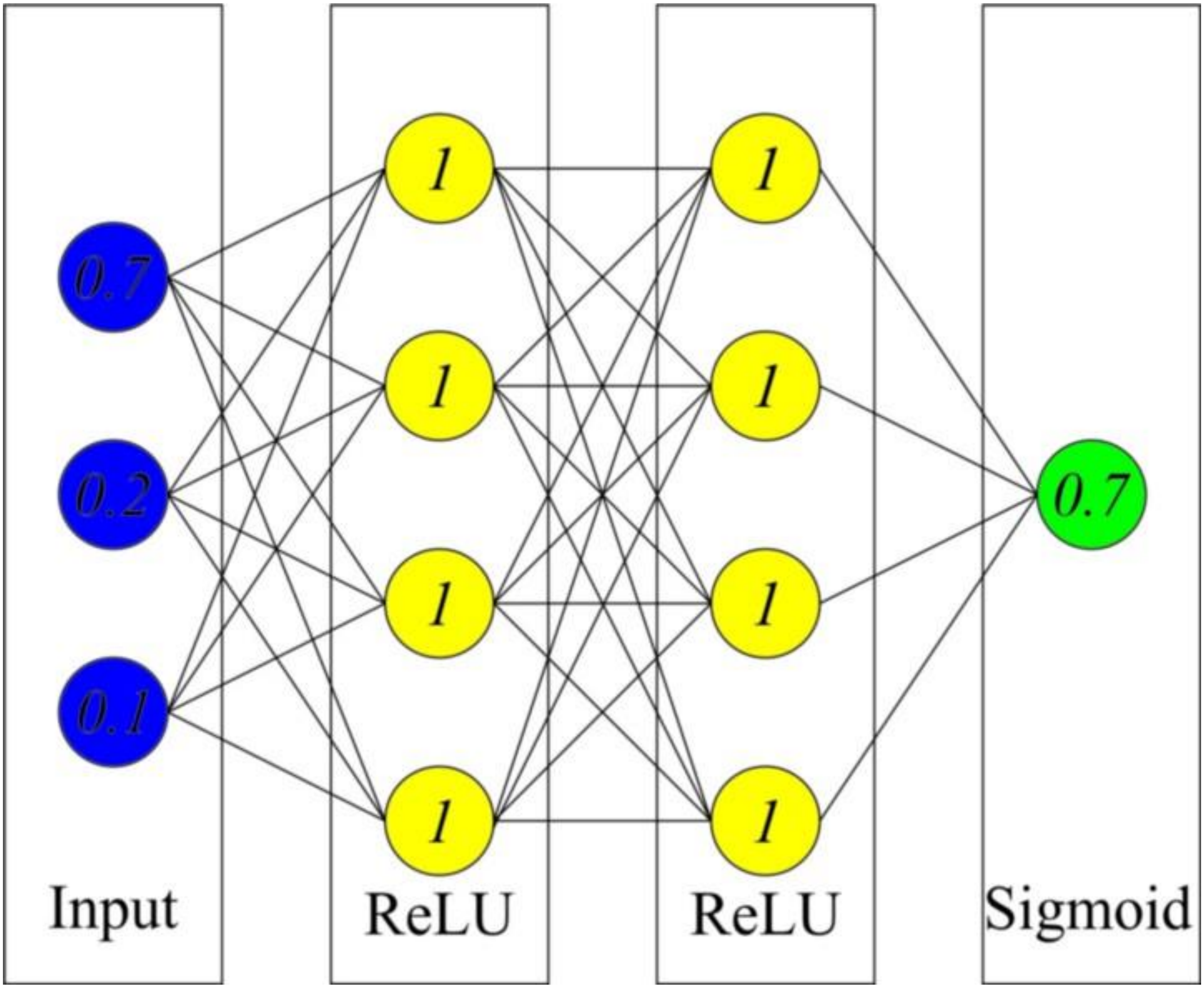
The easiest way to initialize weights is to initialize all of them with bios to zero.

Initializing the weights and biases to zero will surely lead us to the dead neuron problem.

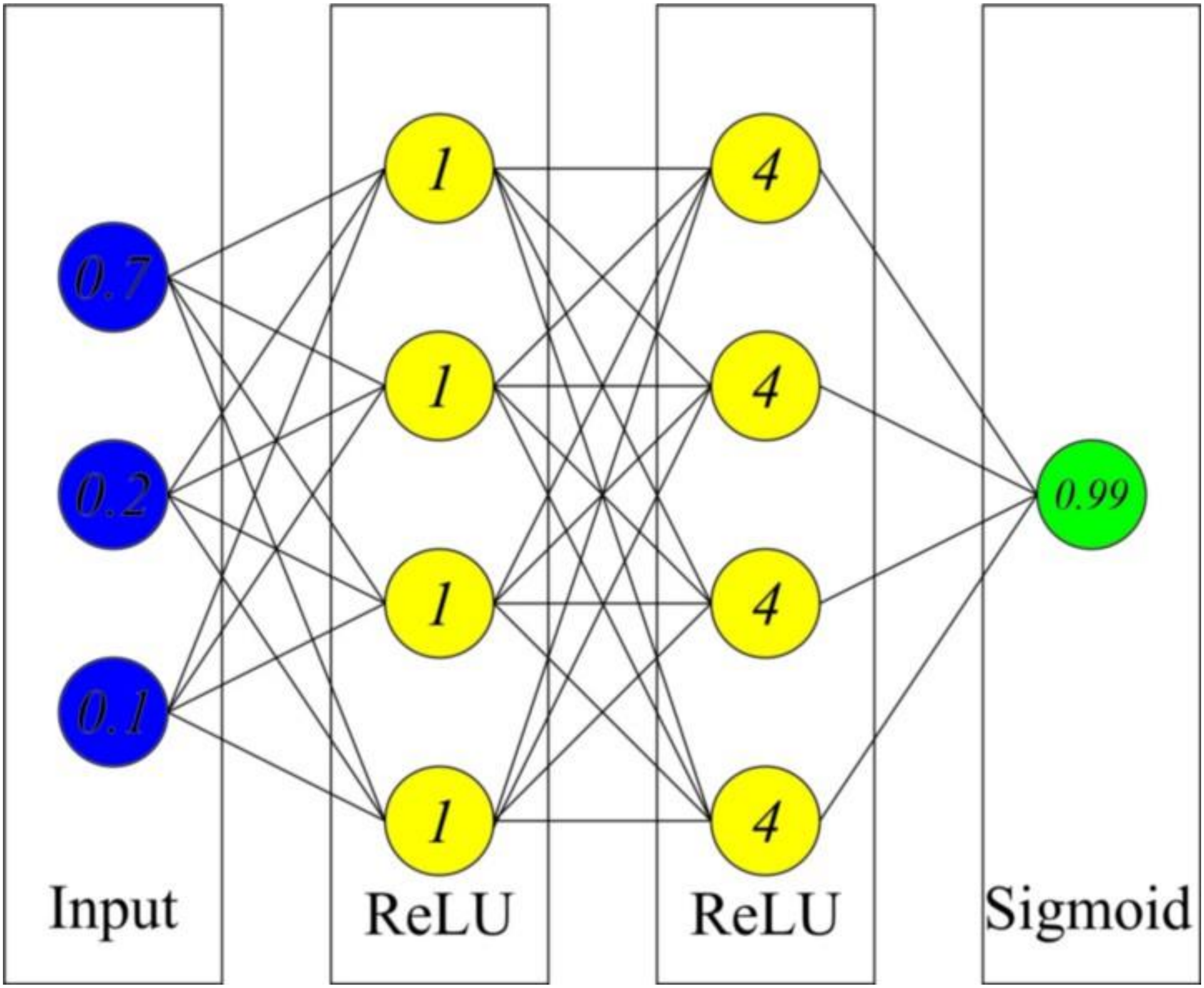


To solve this problem, the bios value of each neuron can be set equal to one. Weights will definitely change. Because Relu neurons will produce non-zero output, but the changes made will be inappropriate. On the other hand, each neuron in the same layer will have the same behavior and the same weight.

This phenomenon is called symmetric problem.



This problem is not solved even by initialization with a fixed value for the weights.



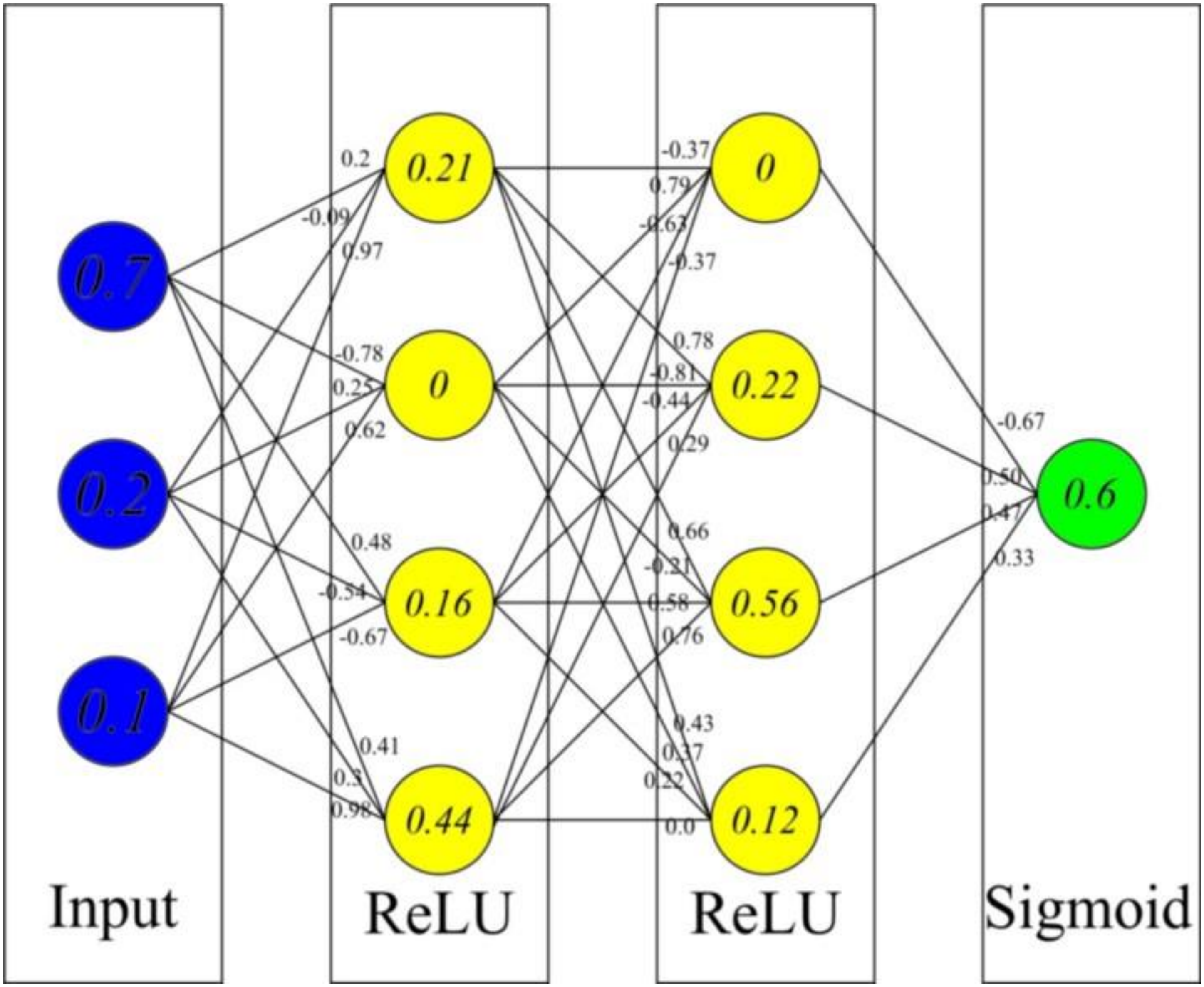
There are several ways to prevent this from happening.

- **Random initialization**
- **Xavier initialization**

Random :

We can use random normal distribution to initialize the weights instead of random numbers.

By setting the mean and standard deviation, we can have a normal distribution in a certain range of numbers.



Xavier :

A better technique for initializing a neural network is to control the output variance. The output of layers created by neurons can be from the same distribution.