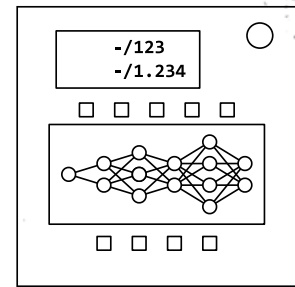


## On the Subject of Perceptron

*Today we diffuse AI around the world. But when it decides to take over this world, we will have to defuse it.*

The module consists of two displays, five buttons above the large display and four buttons below it. The large display shows the perceptron with four hidden layers. Perceptron is a neural network in which each node is connected to each node of the next layer. Input nodes are shown in green, output nodes are shown in red. All other layers are called hidden. The upper display shows the required learning accuracy and the maximum training time (in seconds). The accuracy will be checked after the full training of the perceptron, so even if the required learning accuracy is achieved in less than the maximum available time, the solution will not be valid if the neural network's training time is more than maximum allowed.



To change the number of nodes on the hidden layer, use the buttons at the bottom of the large display. For information about the relationship between layers, hold down the buttons at the top of the large display. The upper displayed value is the convergence rate, the lower is the connection delay. Start training the perceptron by pressing the small display. While the module is in the training stage, the buttons will be inactive. When the perceptron is trained, press the small display. If training time is less than or equals to the maximum allowed and if the required learning accuracy is achieved, the module will be solved. Otherwise you will receive a strike, in this case press the small display again to exit the training stage.

For each pair of adjacent layers, count the number of connections, multiply this value by the corresponding convergence rate and remove the fractional part of the result. Add up the values obtained. The result is learning accuracy.

For each pair of adjacent layers, count the number of connections, multiply this value by the corresponding connection delay and round up. Sum the obtained values. Multiply this sum by total modules count and then add the number of indicators multiplied by 300. The result is training time (in milliseconds).